View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repositorio da Producao Cientifica e Intelectual da Unicamp

UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ELETRÔNICA E MICROELETRÔNICA

DISSERTAÇÃO DE MESTRADO

PROJETO DE UM APLICATIVO PARA SÍNTESE DE FUNÇÕES MULTI-VALORES EM AMBIENTE WINDOWS

MARCO AURÉLIO SELUQUE FREGONEZI

ORIENTADOR: ALBERTO MARTINS JORGE

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Banca Examinadora:

Prof. Dr. Alberto Martins Jorge (Presidente)
Prof. Dr. Elnatan Chagas Ferreira
Prof. Dr. José Antônio Siqueira Dias
Prof. Dr. Nivaldo Vicençotto Serran

27/09/2001

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

F842p

Fregonezi, Marco Aurélio Seluque
Projeto de um aplicativo para síntese de funções multi-valores em ambiente Windows / Marco Aurélio Seluque Fregonezi.--Campinas, SP: [s.n.], 2001.

Orientador: Alberto Martins Jorge.
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Lógica a múltiplos valores. 2. Projeto Auxiliado por computador. 3. Eletrônica digital. I. Jorge, Alberto Martins. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Resumo

Esta dissertação consiste na criação de um programa utilizando a linguagem de programação *BASIC* para ambiente *Windows™* cuja finalidade é a de realizar a síntese de funções lógicas multi-valores (MVL) utilizando a álgebra de Post estendida. A técnica de síntese utilizada é a da separação em sub-funções binárias. Diversas técnicas de minimização são utilizadas, sendo uma delas intuitiva, a da formação de linhas e colunas completas. O resultado da síntese, que consiste na expressão algébrica reduzida, bem como os passos para sua realização, ou seja, as tabelas e expressões relativas às sub-funções binárias podem ser enviadas para um arquivo, para a impressora ou para a área de transferência. Diversas opções de funções e de visualização são disponibilizadas.

Abstract

This work consists in the creation of a program using *BASIC* programming language for *Windows™* of which the finality is to perform multi-valued-logic (MVL) functional synthesis using extended Post algebra. The synthesis technique used is the subdivision of the function into binary sub-functions. Several minimization techniques are used; one of them is intuitive, the complete lines and columns generation. The result of the synthesis, i.e. the reduced algebraic expression, the truth-tables and expressions concerning to the binary sub-functions may be sent to a file, to a printer or to the clipboard. Many function and visualisation options are available.

Agradecimentos

A Deus por ter dado a capacitação necessária e por ter colocado pessoas que tanta importância tiveram na realização desta obra, citadas a seguir:

- Prof. Dr. Alberto Martins Jorge, pela orientação e pela criação da álgebra estendida de Post e da técnica de síntese utilizada, âmago deste trabalho.
- Os meus pais, pelo incentivo e suporte dado.
- Os colegas de pesquisa Alcino José Biazon Filho e Luciana Prado do Nascimento, por ajudas pertinentes à lógica multi-valores.
- Os colegas de sala Fernando e José Carlos, por ajudas genéricas.
- Paulo (*webmaster*) e Fernando Braga, por ajudas pertinentes à linguagem de programação BASIC e questões de *hardware*.
- Ademilde, pelo amparo nas questões burocráticas e protocolares.
- Dona Zilna, pela boa atuação na manutenção do ambiente de trabalho.
- Todos os integrantes do DEMIC (professores, funcionários e alunos), que proporcionaram um ambiente de trabalho descontraído e humano.

Aos meus pais

Lair

Míriam

Índice

Introdução

Quando a álgebra de Boole foi formalizada, em meados do século XIX pelo matemático inglês George Boole (1815-1864) [01,02,03], não se imaginou que tal conhecimento tenha tanta aplicação em circuitos elétricos, fato este que viria a ocorrer muito tempo depois.

Também não se sabia que tais conceitos eram um caso particular de uma álgebra mais abrangente, genérica e poderosa. O matemático indiano Augustus De Morgan (1806 – 1871) [04] realizou grandes avanços na lógica matemática.

Em 1917,o lógico ucraniano Jan Lukasiewicz (1878-1956) desenvolveu o cálculo proposicional de três valores e trabalhou na lógica multi-valores [02,05]. Em 1920, o matemático polonês Emil Leon Post (1897-1954) fez um estudo matemático da lógica de três valores de Lukasiewicz [06,07,08].

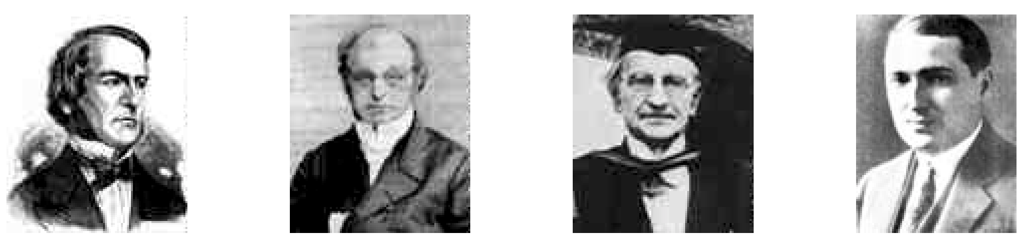


Figura 0.1. George Boole, Augustus De Morgan, Jan Lukasiewicz, Emil Leon Post

O trabalho de Post, em 1921, foi a primeira álgebra publicada com completeza funcional [02,09,10] para qualquer base, introduzindo dois operadores: operador de adição de uma única variável $(1+x)\text{mod}_{BASE}$ e o operador de máximo de duas variáveis $MAX(x_1, x_2)$ [10,11].

A primeira álgebra de n-valores verdadeiramente funcionalmente completa relacionada ao trabalho de Post foi introduzida em 1942 por Rosembloom [08,11].

A álgebra de Boole é um sub-conjunto da álgebra de Post. Esta tem recebido muita atenção não somente dos matemáticos, mas também dos engenheiros [12].

Em qualquer sistema numérico, quanto menor for a base, maior será o número de dígitos necessários para expressar uma dada quantidade [10,11], porém, a complexidade se sistemas, circuitos e manipulações algébricas para síntese de funções cresce à medida que se aumenta a base [11].

A utilização de base 4 torna mais fácil a sua utilização em conjunto com sistemas binários [07,11].

Atualmente, o maior problema apresentado por sistemas binários é o das interconexões internas ao chip e entre chips [07,10,11,13]. Um grande número de interconexões internas ao chip aumenta a área do dispositivo e dificulta o projeto das máscaras do circuito integrado. Um grande número de interconexões entre chips causa problemas de capacitâncias e indutâncias parasitas oriundas das trilhas muito próximas umas das outras (capacitor) e muito longas (indutor).

Uma alternativa para contornar parte destas dificuldades é transmitir mais informação por ligação entre módulos funcionais internos ao chip ou na interligação entre chips [07,11], diminuindo a área do chip [14].

Os circuitos digitais binários são projetados para a operação em dois estados de polarização, isto é, a tensão de saída tem somente dois estados (valores): baixo ou alto.[15].

Nos circuitos digitais multi-valores, este conceito deve ser estendido além dos estados de polarização mencionados, utilizando regiões onde a corrente no semicondutor é mensurável, o que aumenta a potência consumida, mas já existem circuitos MVL que não utilizam tais regiões [16].

Os circuitos digitais multi-valores podem ser construídos de duas formas: modo de corrente e modo de tensão [10,16], dentre outros. O primeiro fornece a vantagem da rapidez de comutação. Por não utilizar apenas dois pontos de polarização, estes circuitos possuem alto consumo de potência em regime não-transitório; mas há projetos nos quais esse problema é resolvido utilizando-se vários tipos de transistores, com tensões de limiar diferentes [16]

Para que os circuitos possam ser construídos, é necessário fazer uma análise profunda da lógica empregada, para que haja otimização na maneira como as portas lógicas são organizadas. Por esse motivo, deve-se fazer as tabelas verdade das funções desejadas e minimizar as funções, obtendo expressões algébricas que impliquem em um circuito mais compacto.

Este trabalho consiste na criação de um software cuja finalidade é a de auxiliar o projeto e análise de circuitos MVL. Sua tarefa é a de fornecer, a partir da tabela verdade fornecida pelo usuário, a expressão algébrica minimizada, por meio da qual o circuito integrado será projetado.

Capítulo 1
Lógica

1.1 Resumo de Lógica Booleana

Variáveis lógicas são aquelas que somente podem assumir estados distintos [01], que, no caso da lógica booleana, são:

- 0
- 1

Esses valores estão atribuídos aos seguintes significados, dependendo da lógica empregada:

- Lógica negativa [01]:
    - 0: Verdadeiro ou ON ou HIGH
    - 1: Falso ou OFF ou LOW
- Lógica positiva [01]
    - 0: Falso ou OFF ou LOW
    - 1: Verdadeiro ou ON ou HIGH

Normalmente, em circuitos elétricos, o valor lógico HIGH é atribuído à tensão de alimentação e o valor lógico LOW ao valor de tensão mínimo, geralmente o *terra*.

Tem-se três operações básicas: AND, OR e NOT [15].

- A operação AND (*e*) também é chamada de <u>conjunção</u> ($\wedge$), <u>intersecção</u> ($\cap$) <u>multiplicação lógica</u> (*) ou mínimo [02,08,10,11,13,14,17,18,19].
- A operação OR (*ou*) também é chamada de <u>disjunção</u> ($\vee$) , <u>união</u> ($\cup$) , <u>adição lógica</u> (+) ou máximo [02,08,10,11,13,14,17,18,19].

- A operação NOT também é chamada de complementação ou negação e é um tipo de deslocador.

Os operadores AND e OR são definidos como segue [16]:

$$A * B = \begin{cases} A & se \quad A \le B \\ B & se \quad B < A \end{cases} \quad (1.1a)$$

$$A + B = \begin{cases} A & se \quad A \ge B \\ B & se \quad B > A \end{cases} \quad (1.1b)$$

OR

+	**0**	**1**
0	0	1
1	1	1

AND

*	**0**	**1**
0	0	0
1	0	1

NOR

+	**0**	**1**
0	1	0
1	0	0

NAND

*	**0**	**1**
0	1	1
1	1	0

NOT

0	**1**
1	0

Tabela 1.1 – Operações da lógica booleana.

A conversão entre os operadores OR e AND é feita por meio das duas fórmulas de De Morgan [01]:

$$\overline{A * B} = \overline{A} + \overline{B} \quad (1.2a)$$
$$\overline{A + B} = \overline{A} * \overline{B} \quad (1.2b)$$

As operações OR e AND são chamadas de conectivos lógicos [17] e são definidas para dois argumentos. Tais operadores obedecem a lei da comutatividade, ou seja, as entradas podem ser comutadas sem alterar a resposta da porta. Além desses, existem outros conectivos:

- **condicional ou implicação**: (→), (se) [02,17], não obedece à lei da comutatividade.
- **bicondicional ou equivalência**: (↔) ou (≡), (se e somente se) [02,17].

Até o presente momento, tais conectivos não têm sido empregados como portas lógicas em circuitos digitais, mas há estudos sobre sua incorporação à lógica MVL [02].

$\overline{A+B}$

	0	**1**
0	1	0
1	0	0

=

$\overline{A}*\overline{B}$

	1	**0**
1	1	0
0	0	0

$\overline{A*B}$

	0	**1**
0	1	1
1	1	0

=

$\overline{A}+\overline{B}$

	1	**0**
1	1	1
0	1	0

Tabela 1.2 – Demonstração das fórmulas de De Morgan.

A fórmulas de De Morgan mostram que este sistema de regras de inferência, constituído das três operações (NOT, OR, AND) não precisa das três operações para ser completo, uma das duas últimas operações pode ser eliminada, sem causar qualquer dano ao sistema lógico. O motivo disso é que a operação OR é o dual da operação AND, ou seja, nenhuma delas acrescenta propriedades ao sistema quando a outra já existir juntamente com a operação NOT. As fórmulas de De Morgan são tautologias [17]

- As operações OR e NAND são derivadas do conjunto {NOT, AND}.
- As operações AND e NOR são derivadas do conjunto {NOT, OR}.

Basta um deslocador e um conectivo para formar um sistema lógico completo.

A função expressa por meio de apenas dois operadores (um deslocador e um conectivo) está na forma canônica [11,20].

Embora a definição das operações OR e AND sejam feitas para duas variáveis, podemos ter, na prática, quantas entradas quisermos, desde que respeitemos as seguintes propriedades:

- Na operação AND, o valor dominante é o **0**.
- Na operação OR, o valor dominante é o **1**.

AND

+	**00**	**01**	**10**	**11**
0	0	0	0	0
1	0	0	0	1

OR

*	**00**	**01**	**10**	**11**
0	0	1	1	1
1	1	1	1	1

Tabela 1.3 – Três entradas

AND

+	**00**	**01**	**10**	**11**
00	0	0	0	0
01	0	0	0	0
10	0	0	0	0
11	0	0	0	1

OR

*	**00**	**01**	**10**	**11**
00	0	1	1	1
01	1	1	1	1
10	1	1	1	1
11	1	1	1	1

Tabela 1.4 – Quatro entradas

Percebe-se, nas tabelas acima, que linhas ou colunas da tabela podem ser associadas a mais de uma variável de entrada. Esta é uma forma de representação bidimensional para funções com mais dimensões. Há, ainda, algumas fórmulas auxiliares [01,15]:

$$A*1=A \qquad (1.3a)$$
$$A+0=A \qquad (1.3b)$$
$$A*0=0 \qquad (1.3c)$$
$$A+1=1 \qquad (1.3d)$$
$$A*\overline{A}=0 \qquad (1.3e)$$
$$A+\overline{A}=1 \qquad (1.3f)$$
$$A*(B+C)=(A*B)+(A*C) \qquad (1.3g)$$
$$A+(B*C)=(A+B)*(A+C) \qquad (1.3h)$$
$$A+(A*B)=A \qquad (1.3i)$$
$$A*(A+B)=A \qquad (1.3j)$$

- O valor 0 é o elemento neutro da operação OR.
- O valor 1 é o elemento neutro da operação AND.

As fórmulas (1.3g-h) consistem na propriedade da distributividade entre as operações OR e AND. As fórmulas (1.3i-j) consistem na lei da absorção binária.

A disponibilidade dos operadores OR e AND juntos permite que se expresse a função na forma de soma de produtos (somatório de mintermos, $\Sigma \min$) ou na forma de produtos de somas (produtório de maxtermos, $\Pi \max$) [01,10,14,15,19], como será visto mais adiante. Este é o motivo de se empregar estas duas operações e não apenas uma delas.

Zeros

0	*A = 0*	*A = 1*
B = 0	$A + B$	$\overline{A} + B$
B = 1	$A + \overline{B}$	$\overline{A} + \overline{B}$

Uns

1	*A = 0*	*A = 1*
B = 0	$\overline{A} * \overline{B}$	$A * \overline{B}$
B = 1	$\overline{A} * B$	$A * B$

Tabela 1.5

$$0 = (A + B) * (\overline{A} + B) * (A + \overline{B}) * (\overline{A} + \overline{B}) \quad (1.4a)$$
$$1 = (\overline{A} * \overline{B}) + (A * \overline{B}) + (\overline{A} * B) + (A * B) \quad (1.4b)$$

1.2 Lógica de três valores de Lukasiewicz e sua expansão para mais valores

Os conceitos da lógica binária também valem para mais valores. A simbologia utilizada por Lukasiewicz para expressar os conectivos (*) e (+) é ($\wedge$) e ($\vee$), respectivamente [02]. A proposta de Lukasiewicz foi a de três valores, mas, este conceito pode ser expandido para mais valores. A nomenclatura dos valores era 0, ½ e 1 [02], mas, para facilitar a implementação para quatro valores, serão utilizados números inteiros.

A	**¬A**
0	1
1	0

$\wedge$	**0**	**1**
0	0	0
1	0	1

$\vee$	**0**	**1**
0	0	1
1	1	1

A	**¬A**
0	2
1	1
2	0

$\wedge$	**0**	**1**	**2**
0	0	0	0
1	0	1	1
2	0	1	2

$\vee$	**0**	**1**	**2**
0	0	1	2
1	1	1	2
2	2	2	2

A	**¬A**
0	3
1	2
2	1
3	0

$\wedge$	**0**	**1**	**2**	**2**
0	0	0	0	0
1	0	1	1	1
2	0	1	2	2
3	0	1	2	3

$\vee$	**0**	**1**	**2**	**2**
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

Tabela 1.6 – Dois, três e quatro valores, respectivamente.

Neste caso, o operador complemento também pode ser chamado de inversor (¬)

Os dois conectivos lógicos podem ser formalizados como segue [16]:

$$\min(A,B)=\begin{cases} A \text{ se } A \le B \\ B \text{ se } A > B \end{cases} \quad (1.5a)$$

$$\max(A,B)=\begin{cases} A \text{ se } A \ge B \\ B \text{ se } A < B \end{cases} \quad (1.5b)$$

As fórmulas de De Morgan continuam valendo:

$$\neg(A \wedge B) = (\neg A) \vee (\neg B) \quad (1.6a)$$

$$\neg(A \vee B) = (\neg A) \wedge (\neg B) \quad (1.6b)$$

Também é válida a propriedade distributiva:

$$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C) \quad (1.7a)$$

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C) \quad (1.7b)$$

Este sistema lógico possui a vantagem de se enquadrar na propriedade da dualidade dos circuitos elétricos; o operador mínimo é o dual do máximo. Sua desvantagem consiste na não completeza do sistema.

1.3 Lógica de Post e sua expansão

Em um sistema formal, as regras de formação das expressões lógicas implicam no uso de [17]:

- Deslocadores (operadores de um argumento)
- Conectivos (operadores de dois argumentos)
- Quantificadores

O uso dos quantificadores (universal e existencial) [17] e dos conectivos condicional e bicondicional [17] está além do escopo deste trabalho, mas já existem trabalhos neste sentido [02].

1.3.1- Deslocadores

Nesta lógica, são definidas duas operações de um argumento (*n* é o valor da base adotada), chamadas de deslocadores (*shifters*):

- *Clockwise cycle* ou deslocamento para a direita ou rotação cíclica unária ou negação cíclica de Post, aqui chamada de **TOPO**. [07,09,11,13,20,22]
- *Counterclockwise cycle* ou deslocamento para a esquerda, aqui chamado de **BASE**. [07,09,11,13,20,22]

A nomenclatura TOPO e BASE não fazem parte da formulação original de Post. Suas definições são mostradas abaixo, respectivamente [09,11,13,20].

$$\boxed{\overline{A} = \begin{cases} A+1 & se \quad A < n-1 \\ 0 & se \quad A = n-1 \end{cases} \quad (1.8a)}$$

$$\boxed{\underline{A} = \begin{cases} A-1 & se \quad A > 0 \\ n-1 & se \quad A = 0 \end{cases} \quad (1.8b)}$$

Os deslocadores podem ser implementado computacionalmente da seguinte forma:

$$\overline{A} = (A+1) \quad \text{mod} \quad n \qquad (1.9a)$$
$$\underline{A} = (A+n-1) \quad \text{mod} \quad n \qquad (1.9b)$$

A_2	$\overline{A}_2$	$\underline{A}_2$
0	1	1
1	0	0

A_3	$\overline{A}_3$	$\underline{A}_3$
0	1	2
1	2	0
2	0	1

A_4	$\overline{A}_4$	$\underline{A}_4$
0	1	3
1	2	0
2	3	1
3	0	2

Tabela 1.7 [11]

Estas operações também podem ser definidas para deslocamentos múltiplos [07], mas serão analisadas apenas a operação TOPO e BASE de um deslocamento. Pode-se ter múltiplas operações TOPO e BASE, obtendo, desta forma deslocamentos múltiplos.

Deve-se tomar cuidado para não confundir a palavra BASE referente ao deslocador com a palavra BASE referente à quantidade de valores (dimensão).

Uma atenção especial deve ser dada quanto à simbologia adotada relativa ao número de valores empregado. Os traços acima e abaixo da letra, representando um ou mais deslocamentos para a esquerda (BASE) ou para a direita (TOPO), têm seu significado alterado dependendo da base adotada.

Em dois valores, temos:

$$\overline{\overline{A}}_2 = A_2 \quad (1.10a)$$
$$\overline{A_2} = \underline{A_2} \quad (1.10b)$$

Em três valores, temos:

$$\overline{\overline{\overline{A}}}_3 = A_3 \quad (1.10c)$$
$$\overline{\overline{A}}_3 = \underline{A_3} \quad (1.10d)$$
$$\underline{\underline{A}}_3 = \overline{A}_3 \quad (1.10e)$$

Em quatro valores, temos:

$$\overline{\overline{A}}_4 = \underline{\underline{A}}_4 \quad (1.10f)$$

1.3.2- Conectivos

São, ao todo, quatro conectivos elementares.

A	***B***	$\wedge$	$\vee$	$\rightarrow$	$\equiv$
0	**0**	0	0	1	1
0	**1**	0	1	1	0
1	**0**	0	1	0	0
1	**1**	1	1	1	1

Tabela 1.8 – Tabelas-verdade para os conectivos booleanos [02].

Os conectivos utilizados na lógica de Post estendida, ou seja, operadores de dois argumentos são definidos a partir das operações MÍINIMO e TOPO, que são considerados primitivos [11]. Na expansão para mais operadores, o conectivo MÍNIMO recebe o nome de “ALFA” [20].

$$\boxed{A\alpha B = \begin{cases} A & se \quad A \le B \\ B & se \quad B < A \end{cases} \quad (1.11)}$$

Outros conectivos são definidos com o objetivo de simplificar a notação [20]:

$$\overline{x\,\alpha\,y} = \overline{x}\,\overline{\alpha}\,\overline{y} = \overline{x}\,\beta\,\overline{y} \quad (1.12a)$$
$$\overline{x\,\beta\,y} = \overline{x}\,\overline{\beta}\,\overline{y} = \overline{x}\,\gamma\,\overline{y} \quad (1.12b) \quad [11,20,21]$$
$$\overline{x\,\gamma\,y} = \overline{x}\,\overline{\gamma}\,\overline{y} = \overline{x}\,\alpha\,\overline{y} \quad (1.12c)$$

$$\beta \equiv \overline{\alpha} \quad (1.13a)$$
$$\gamma \equiv \overline{\beta} \quad (1.13b)$$
$$\delta \equiv \overline{\gamma} \quad (1.13c)$$

e assim por diante. Pode-se, por meio deste processo, obter tantos operadores quanto se queira; no entanto, somente alguns são relevantes, de acordo com a base adotada.

α	**0**	**1**	**2**
0	0	0	0
1	0	1	1
2	0	1	2

Tabela 1.9 – Tabelas-verdade para o operador ALFA de três valores [21].

Por meio desta tabela, obtém-se as tabelas para os demais operadores.

$\overline{x \alpha y}$

$\overline{\alpha}$	**0**	**1**	**2**
0	1	1	1
1	1	2	2
2	1	2	0

=

$\overline{x}\, \beta\, \overline{y}$

β	**1**	**2**	**0**
1	1	1	1
2	1	2	2
0	1	2	0

$\rightarrow$

$x \beta y$

β	**0**	**1**	**2**
0	0	1	2
1	1	1	1
2	2	1	2

$\overline{x \beta y}$

$\overline{\beta}$	**0**	**1**	**2**
0	1	2	0
1	2	2	2
2	0	2	0

=

$\overline{x}\, \gamma\, \overline{y}$

γ	**1**	**2**	**0**
1	1	2	0
2	2	2	2
0	0	2	0

$\rightarrow$

$x \gamma y$

γ	**0**	**1**	**2**
0	0	0	2
1	0	1	2
2	2	2	2

$\overline{x \gamma y}$

$\overline{\gamma}$	**0**	**1**	**2**
0	1	1	0
1	1	2	0
2	0	0	0

=

$\overline{x}\, \delta\, \overline{y}$

δ	**1**	**2**	**0**
1	1	1	0
2	1	2	0
0	0	0	0

$\rightarrow$

$x \delta y \equiv x \alpha y$

α	**0**	**1**	**2**
0	0	0	0
1	0	1	1
2	0	1	2

Tabela 1.10

O mesmo procedimento pode ser utilizado para quatro ou mais valores.

Também pode ser utilizado o deslocador BASE para a obtenção daquelas tabelas.

$\underline{x\,\alpha\,y}$

$\underline{\alpha}$	**0**	**1**	**2**
0	2	2	2
1	2	0	0
2	2	0	1

=

$\underline{x}\,\gamma\,\underline{y}$

γ	**2**	**0**	**1**
2	2	2	2
0	2	0	0
1	2	0	1

$\rightarrow$

$x\,\gamma\,y$

γ	**0**	**1**	**2**
0	0	0	2
1	0	1	2
2	2	2	2

$\underline{x\,\gamma\,y}$

$\underline{\gamma}$	**0**	**1**	**2**
0	2	2	1
1	2	0	1
2	1	1	1

=

$\underline{x}\,\beta\,\underline{y}$

β	**2**	**0**	**1**
2	2	2	1
0	2	0	1
1	1	1	1

$\rightarrow$

$x\,\beta\,y$

β	**0**	**1**	**2**
0	0	1	2
1	1	1	1
2	2	1	2

$\underline{x\,\beta\,y}$

β	**0**	**1**	**2**
0	2	0	1
1	0	0	0
2	1	0	1

=

$\underline{x}\,\alpha\,\underline{y}$

β	**2**	**0**	**1**
2	2	0	1
0	0	0	0
1	1	0	1

$\rightarrow$

$x\,\alpha\,y$

β	**0**	**1**	**2**
0	0	0	0
1	0	1	1
2	0	1	2

Tabela 1.11

Sendo assim, os conectivos são definidos como [13,20,21,23]:

α	**0**	**1**	**2**
0	0	0	0
1	0	1	1
2	0	1	2

β	**0**	**1**	**2**
0	0	1	2
1	1	1	1
2	2	1	2

γ	**0**	**1**	**2**
0	0	0	2
1	0	1	2
2	2	2	2

Tabela 1.12 – Tabelas-verdade para os três conectivos de três valores.

Os conectivos para quatro valores podem ser obtidos da mesma forma [09]:

α	**0**	**1**	**2**	**3**
0	0	0	0	0
1	0	1	1	1
2	0	1	2	2
3	0	1	2	3

β	**0**	**1**	**2**	**3**
0	0	1	2	3
1	1	1	1	1
2	2	1	2	2
3	3	1	2	3

γ	**0**	**1**	**2**	**3**
0	0	0	2	3
1	0	1	2	3
2	2	2	2	2
3	3	3	2	3

δ	**0**	**1**	**2**	**3**
0	0	0	0	3
1	0	1	1	3
2	0	1	2	3
3	3	3	3	3

Tabela 1.13

De uma maneira geral, a função ALFA pode ser mostrada como na tabela abaixo:

α	0	1	2	...
0	0 mod n	0 mod n	0 mod n	...
1	0 mod n	1 mod n	1 mod n	...
2	0 mod n	1 mod n	2 mod n	...
...	...	...	...	...

Tabela 1.14 – Tabelas-verdade para o operador ALFA de n valores.

Uma vez que $\overline{\overline{A}}_2 = A_2 \ (1.6a)$, tem-se que $\begin{matrix} \gamma_2 \equiv \alpha_2 & (1.14a) \\ \delta_2 \equiv \beta_2 & (1.14b) \end{matrix}$ [11].

Uma vez que $\overline{\overline{\overline{A}}}_3 = A_3 \ (1.7a)$, tem-se que $\delta_3 \equiv \alpha_3 \ (1.14c)$.

$$\begin{matrix} \alpha_2 \equiv \overline{\beta_2} & (1.15a) \\ \beta_2 \equiv \overline{\alpha_2} & (1.15b) \end{matrix} \qquad \begin{matrix} \underline{\beta_3} \equiv \alpha_3 \equiv \overline{\gamma_3} & (1.15c) \\ \underline{\gamma_3} \equiv \beta_3 \equiv \overline{\alpha_3} & (1.15d) \\ \underline{\alpha_3} \equiv \gamma_3 \equiv \overline{\beta_3} & (1.15e) \end{matrix} \qquad \begin{matrix} \underline{\beta_4} \equiv \alpha_4 \equiv \overline{\delta_4} & (1.15f) \\ \underline{\gamma_4} \equiv \beta_4 \equiv \overline{\alpha_4} & (1.15g) \\ \underline{\delta_4} \equiv \gamma_4 \equiv \overline{\beta_4} & (1.15h) \\ \underline{\alpha_4} \equiv \delta_4 \equiv \overline{\gamma_4} & (1.15i) \end{matrix}$$

Tem-se, então, as extensões da lei de De Morgan para lógica multi-valores:

Dois valores

$$\overline{A\alpha B} = \overline{A}\beta\,\overline{B} \quad (1.16a)$$
$$\overline{A\beta B} = \overline{A}\alpha\,\overline{B} \quad (1.16b)$$

Três valores

$$\begin{matrix} \overline{A\alpha B} = \overline{A}\beta\,\overline{B} & (1.16c) & \qquad \underline{A\alpha B} = \underline{A}\gamma\,\underline{B} & (1.16f) \\ \overline{A\beta B} = \overline{A}\gamma\,\overline{B} & (1.16d) & \qquad \underline{A\beta B} = \underline{A}\alpha\,\underline{B} & (1.16g) \\ \overline{A\gamma B} = \overline{A}\alpha\,\overline{B} & (1.16e) & \qquad \underline{A\gamma B} = \underline{A}\beta\,\underline{B} & (1.16h) \end{matrix}$$

Quatro valores

$$\begin{matrix} \overline{A\alpha B} = \overline{A}\beta\,\overline{B} & (1.16i) & \overline{\overline{A\alpha B}} = \overline{\overline{A}}\gamma\,\overline{\overline{B}} & (1.16m) & \underline{A\alpha B} = \underline{A}\delta\,\underline{B} & (1.16q) \\ \overline{A\beta B} = \overline{A}\gamma\,\overline{B} & (1.16j) & \overline{\overline{A\beta B}} = \overline{\overline{A}}\delta\,\overline{\overline{B}} & (1.16n) & \underline{A\beta B} = \underline{A}\alpha\,\underline{B} & (1.16r) \\ \overline{A\gamma B} = \overline{A}\delta\,\overline{B} & (1.16k) & \overline{\overline{A\gamma B}} = \overline{\overline{A}}\alpha\,\overline{\overline{B}} & (1.16o) & \underline{A\gamma B} = \underline{A}\beta\,\underline{B} & (1.16s) \\ \overline{A\delta B} = \overline{A}\alpha\,\overline{B} & (1.16l) & \overline{\overline{A\delta B}} = \overline{\overline{A}}\beta\,\overline{\overline{B}} & (1.16p) & \underline{A\delta B} = \underline{A}\gamma\,\underline{B} & (1.16t) \end{matrix}$$

Estas 20 fórmulas são demonstradas no apêndice um.

Percebe-se que, a partir de um único conectivo juntamente com um deslocador, pode-se obter todas as demais operações. A completeza e a correção do sistema é mantida com apenas um conectivo e um deslocador, qualquer que seja a base adotada. Tais conceitos são explicados a seguir:

- **Teoria da Correção** (*soundness*): "Um sistema de regras de inferência é correto se toda conclusão derivada de seu uso é conseqüência lógica das premissas a partir das quais é obtida." [17]
- **Teoria da Consistência** (*consistency*): "Um sistema de regras de inferência é consistente se não existe nenhuma sentença tal que ela e sua negação sejam deriváveis a partir do conjunto vazio de premissas." Se um sistema é correto, então ele é consistente, mas a recíproca não é verdadeira. [17]
- **Teoria da Completeza** (*completeness*): "Um sistema de regras de inferência é completo se e somente se é possível derivar, a partir de um conjunto de sentenças, todas as conseqüências daquele conjunto".[17]

Post, ao criar seu sistema lógico formal, poderia ter escolhido qualquer deslocador e qualquer conectivo. Sua escolha foi pelo deslocador TOPO e pelo conectivo ALFA, que, na ocasião, eram chamados de *deslocamento para a direita* e *mínimo*, respectivamente.

Na tabela a seguir, tem-se quatro exemplos de possibilidades de escolha para os operadores. Cada escolha gera um sistema formal totalmente novo, incompatível com os demais.

Todas estas opções têm como intersecção a lógica de dois valores, que é a mesma para todos.

Opção	Deslocador	Conectivo
1_{POST}	Para direita	Mínimo
2	Para direita	Máximo
3	Para esquerda	Mínimo
4	Para esquerda	Máximo

Tabela 1.15

1	2	3	4
$\alpha \equiv \wedge \quad (1.17a)$	$\alpha \equiv \vee \quad (1.17e)$	$\alpha \equiv \wedge \quad (1.17i)$	$\alpha \equiv \vee \quad (1.17m)$
$\beta \equiv \overline{\alpha} \quad (1.17b)$	$\beta \equiv \overline{\alpha} \quad (1.17f)$	$\beta \equiv \underline{\alpha} \quad (1.17j)$	$\beta \equiv \underline{\alpha} \quad (1.17n)$
$\gamma \equiv \overline{\beta} \quad (1.17c)$	$\gamma \equiv \overline{\beta} \quad (1.17g)$	$\gamma \equiv \underline{\beta} \quad (1.17k)$	$\gamma \equiv \underline{\beta} \quad (1.17o)$
$\delta \equiv \overline{\gamma} \quad (1.17d)$	$\delta \equiv \overline{\gamma} \quad (1.17h)$	$\delta \equiv \underline{\gamma} \quad (1.17l)$	$\delta \equiv \underline{\gamma} \quad (1.17p)$

Tabela 1.16

A lógica a ser utilizada é a de Post, com a nomenclatura extendida (TOPO, BASE, ALFA, BETA, etc).

Embora o rigor matemático restrinja os conectivos a apenas dois argumentos, serão vistas, também, expansões destes conceitos para mais argumentos (ou variáveis, ou entradas).

1.4 Considerações gerais

A simbologia das portas utilizadas em circuitos digitais é mostrada abaixo:

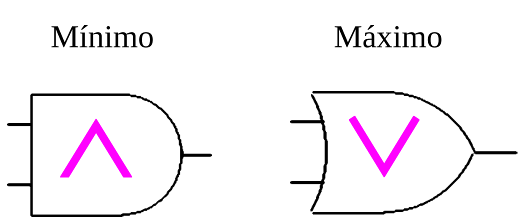


Figura 1.1 – Simbologia [01,15].

A quantidade de possibilidades para funções de n valores e m entradas é dada pela fórmula a seguir:

$$x = n^{n^m} \quad (1.18)\ [08]$$

	m=1	m=2	m=3
n=2	$2^{2^1} = 4$	$2^{2^2} = 16$	$2^{2^3} = 256$
n=3	$3^{3^1} = 27$	$3^{3^2} = 19683$	$3^{3^3} \cong 8*10^{12}$
n=4	$4^{4^1} = 256$	$4^{4^2} \cong 4*10^{9}$	$4^{4^3} \cong 3*10^{38}$

Tabela 1.17

Existe uma hierarquia, entre os valores utilizados na base empregada, para os operadores ALFA, BETA, GAMA e DELTA, como foi visto na página 15. O nome adotado para as posições hierárquicas ou de dominâncias é mostrado a seguir.

Dominante: Corresponde ao valor de maior dominância, ou seja, sempre que uma das variáveis de entrada receber este valor, a variável de saída receberá tal valor (sem levar em consideração algum deslocamento da saída implícito à função, por exemplo, $\overline{\alpha}$).

Secundário: É o valor de segunda dominância, ou seja, sempre que uma das variáveis de entrada receber tal valor, a variável de saída receberá tal valor, exceto se o outro valor for o dominante.

Terciário: Obedece à mesma idéia do secundário, tendo, porém, menor dominância do que a daquele valor.

Indiferente: É o valor de menor dominância. Recebe esse nome porque sempre que houver mais algum outro valor nas entradas, este outro valor prevalecerá. A variável de saída recebe o valor indiferente se e somente se todas as variáveis de entrada receberem este valor.

A dominância do valor indiferente depende da quantidade de valores empregados.

Valores	**Dominância do valor indiferente**
Dois	Secundário
Três	Terciário
Quatro	Quaternário

Tabela 1.18

A lógica multi-valores empregada é ciclicamente fechada ou circularmente conectada [09,07], ou seja, um deslocamento do valor máximo da base a dotada para a direita fornece o valor mínimo, enquanto que um deslocamento para a esquerda do valor mínimo fornece o valor máximo da base adotada.

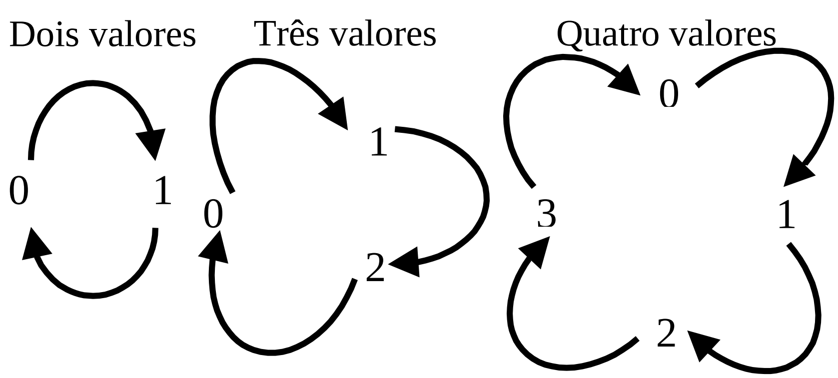


Figura 1.2 – Deslocamento TOPO.

Capítulo 2
Características da Lógica MVL

2.1 Características comuns

Algumas propriedades são comuns a qualquer quantidade de valores:

1 Operação com o indiferente:

Uma variável operada com o valor indiferente do conectivo fornece a própria variável [11].

2 Operação com o dominante:

Uma variável operada com o valor dominante do conectivo fornece valor o dominante [11].

3 Obtenção do dominante:

Uma variável operada com todas as possibilidades de deslocamento fornece o valor dominante do conectivo empregado [11].

4 Distributividade:

Se $\mathrm{op2} = \overline{\mathrm{op1}}$, então $A\,\mathrm{op1}\left(B\,\mathrm{op2}\,C\right) = \left(A\,\mathrm{op1}\,B\right)\mathrm{op2}\left(A\,\mathrm{op1}\,C\right)$ $(2.1a)$[11]

Se $\mathrm{op2} \neq \overline{\mathrm{op1}}$, então $A\,\mathrm{op1}\left(B\,\mathrm{op2}\,C\right) \neq \left(A\,\mathrm{op1}\,B\right)\mathrm{op2}\left(A\,\mathrm{op1}\,C\right)$ $(2.1b)$

2.2 Lógica de Dois Valores

Muito embora a lógica de 2 valores coincida com a lógica binária (lógica de Boole) [11], sua nomenclatura é diferente da daquela, pois é um caso particular de uma lógica mais complexa.

- São definidos dois valores: 0,1.
- São definidos dois conectivos: α **(alfa)** , β **(beta)**.
- É definido um deslocador: **topo**.

- Na operação α, o valor “0” é o dominante e o valor “1” é o indiferente.
- Na operação β, o valor “1” é o dominante e o valor “0” é o indiferente.

α	**0**	**1**
0	0	0
1	0	1

β	**1**	**0**
1	1	1
0	1	0

Tabela 2.1 – Conectivos de dois valores.

- Em dois valores, a operação α também é chamada de AND (*) e a operação $\overline{\alpha}$ também é chamada de NAND.
- Em dois valores, a operação β também é chamada de OR (+) e a operação $\overline{\beta}$ também é chamada de NOR.
- O valor hachurado é o indiferente, pois possui o menor peso.

Percebe-se que, para dois valores, $\beta_2 \equiv \vee$.

Binário	Dois valores
AND (*)	α
OR (+)	β

Tabela 2.2 – Comparação binário/dois valores.

Fórmulas auxiliares:

$$\text{Operação com o indiferente}\begin{cases} A\alpha 1 = A & (2.2a) \\ A\beta 0 = A & (2.2b) \end{cases}$$

$$\text{Operação com o dominante}\begin{cases} A\alpha 0 = 0 & (2.2c) \\ A\beta 1 = 1 & (2.2d) \end{cases}$$

$$\text{Obtenção do dominante}\begin{cases} A\alpha \overline{A} = 0 & (2.2e) \\ A\beta \overline{A} = 1 & (2.2f) \end{cases}$$

$$\text{Distributividade}\begin{cases} A\alpha(B\beta C) = (A\alpha B)\beta(A\alpha C) & (2.2g) \\ A\beta(B\alpha C) = (A\beta B)\alpha(A\beta C) & (2.2h) \end{cases}$$

$$\text{Absorção}\begin{cases} A\beta(A\alpha B) = A & (2.2i) \\ A\alpha(A\beta B) = A & (2.2j) \end{cases}$$

As fórmulas 2.2 correspondem às fórmulas 1.3 substituindo os símbolos “*” e “+” por “α” e “β” respectivamente.

2.3 Lógica de três valores

- São definidos três valores: 0,1 e 2.
- São definidos três conectivos: $\boldsymbol{\alpha}$ **(alfa)**, $\boldsymbol{\beta}$ **(beta)** e $\boldsymbol{\gamma}$ **(gama)**.
- São definidos dois deslocadores: **topo** e **base**.

	α	β	γ
Dominante	0	1	2
Secundário	1	2	0
Indiferente	2	0	1

Tabela 2.3 – Hierarquia de dominâncias para os conectivos de três valores.

$\boldsymbol{\alpha}$	**0**	**1**	**2**
0	0	0	0
1	0	1	1
2	0	1	2

$\boldsymbol{\beta}$	**1**	**2**	**0**
1	1	1	1
2	1	2	2
0	1	2	0

$\boldsymbol{\gamma}$	**2**	**0**	**1**
2	2	2	2
0	2	0	0
1	2	0	1

Tabela 2.4 – Conectivos de três valores.

Fórmulas auxiliares:

Operação com o indiferente

$$A\alpha 2 = A \ \ (2.3a)$$
$$A\beta 0 = A \ \ (2.3b)$$
$$A\gamma 1 = A \ \ (2.3c)$$

Operação com o dominante

$$A\alpha 0 = 0 \ \ (2.3d)$$
$$A\beta 1 = 1 \ \ (2.3e)$$
$$A\gamma 2 = 2 \ \ (2.3f)$$

Obtenção do dominante

$$A\alpha\overline{A}\alpha\underline{A} = 0 \ \ (2.2g)$$
$$A\beta\overline{A}\beta\underline{A} = 1 \ \ (2.2h)$$
$$A\gamma\overline{A}\gamma\underline{A} = 2 \ \ (2.2i)$$

Distributividade Crescente [11,12]

$$A\alpha(B\beta C) = (A\alpha B)\beta(A\alpha C) \ \ (2.3j)$$
$$A\beta(B\alpha C) = (A\beta B)\gamma(A\beta C) \ \ (2.3k)$$
$$A\gamma(B\alpha C) = (A\gamma B)\alpha(A\gamma C) \ \ (2.3l)$$

Não-Distributividade Decrescente [11]

$$A\alpha(B\gamma C) \neq (A\alpha B)\gamma(A\alpha C) \ \ (2.3m)$$
$$A\beta(B\alpha C) \neq (A\beta B)\alpha(A\beta C) \ \ (2.3n)$$
$$A\gamma(B\beta C) \neq (A\gamma B)\beta(A\gamma C) \ \ (2.3o)$$

Absorção Crescente

$$A\alpha(A\beta(A\gamma B)) = A \ \ (2.3p)$$
$$A\beta(A\gamma(A\alpha B)) = A \ \ (2.3q)$$
$$A\gamma(A\alpha(A\beta B)) = A \ \ (2.3r)$$

Absorção Decrescente

$$A\alpha(A\gamma(A\beta B)) = A \ \ (2.3s)$$
$$A\beta(A\alpha(A\gamma B)) = A \ \ (2.3t)$$
$$A\gamma(A\beta(A\alpha B)) = A \ \ (2.3u)$$

A veracidade da distributividade crescente é comprovada pelo método exaustivo, ou seja, construindo-se a tabela-verdade para todas as 27 possibilidades de entrada [11]. A distributividade decrescente fornece desigualdade somente em duas combinações de entrada, cada uma [11].

A	B	C		
1	0 2	2 0	$(A\alpha B)\gamma(A\alpha C)=0$	$A\alpha(B\gamma C)=1$
2	1 0	0 1	$(A\beta B)\alpha(A\beta C)=1$	$A\beta(B\alpha C)=2$
0	1 2	2 1	$(A\gamma B)\beta(A\gamma C)=2$	$A\gamma(B\beta C)=0$

Tabela 2.5 – Desigualdades na distributividade decrescente.

A seguir, temos algumas transformações válidas:

$$(A\alpha B)\beta(A\alpha\overline{B})\beta(A\alpha\underline{B}) = A\alpha(B\beta\overline{B}\beta\underline{B}) = A\alpha 1 \quad (2.4a)$$
$$(A\beta B)\gamma(A\beta\overline{B})\gamma(A\beta\underline{B}) = A\beta(B\gamma\overline{B}\gamma\underline{B}) = A\beta 2 \quad (2.4b)$$
$$(A\gamma B)\beta(A\gamma\overline{B})\beta(A\gamma\underline{B}) = A\gamma(B\alpha\overline{B}\alpha\underline{B}) = A\gamma 0 \quad (2.4c)$$

$$(A\beta B)\alpha(A\beta\overline{B})\alpha(A\beta\underline{B}) = (A\alpha B)\beta(A\alpha\overline{B})\beta(A\alpha\underline{B}) \quad (2.4d)$$
$$(A\gamma B)\beta(A\gamma\overline{B})\beta(A\gamma\underline{B}) = (A\beta B)\gamma(A\beta\overline{B})\gamma(A\beta\underline{B}) \quad (2.4e)$$
$$(A\alpha B)\gamma(A\alpha\overline{B})\gamma(A\alpha\underline{B}) = (A\gamma B)\beta(A\gamma\overline{B})\beta(A\gamma\underline{B}) \quad (2.4f)$$

$$A\beta(B\alpha\overline{B}\alpha\underline{B}) = A\beta 0 = A \quad (2.4g)$$
$$A\gamma(B\beta\overline{B}\beta\underline{B}) = A\gamma 1 = A \quad (2.4h)$$
$$A\alpha(B\gamma\overline{B}\gamma\underline{B}) = A\alpha 2 = A \quad (2.4i)$$

As fórmulas (2.4a-c) são chamadas de conversão para binário, pois eliminam o valor indiferente da operação, restando apenas dois valores. Também correspondem às simplificações para linha ou coluna completa em funções de duas entradas $C=C(A,B)$, como será visto mais adiante.

2.4 Lógica de quatro valores

- São definidos quatro valores: 0,1,2 e 3.
- São definidos quatro conectivos: α **(alfa)**, β **(beta),** γ **(gama)** e δ **(delta)**.
- São definidos três deslocadores: **topo**, **duplo topo** e **base**.

	α	β	γ	δ
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	2	3	0	1
Indiferente	3	0	1	2

Tabela 2.6 – Hierarquia de dominâncias para os conectivos de quatro valores.

α	**0**	**1**	**2**	**3**
0	0	0	0	0
1	0	1	1	1
2	0	1	2	2
3	0	1	2	3

β	**1**	**2**	**3**	**0**
1	1	1	1	1
2	1	2	2	2
3	1	2	3	3
0	1	2	3	0

γ	**2**	**3**	**0**	**1**
2	2	2	2	2
3	2	3	3	3
0	2	3	0	0
1	2	3	0	1

δ	**3**	**0**	**1**	**2**
3	3	3	3	3
0	3	0	0	0
1	3	0	1	1
2	3	0	1	2

Tabela 2.7 – Conectivos de quatro valores.

Fórmulas auxiliares:

Operação com o indiferente

$$A\alpha 3 = A \quad (2.5a)$$
$$A\beta 0 = A \quad (2.5b)$$
$$A\gamma 1 = A \quad (2.5c)$$
$$A\delta 2 = A \quad (2.5d)$$

Operação com o dominante

$$A\alpha 0 = 0 \quad (2.5e)$$
$$A\beta 1 = 1 \quad (2.5f)$$
$$A\gamma 2 = 2 \quad (2.5g)$$
$$A\delta 3 = 3 \quad (2.5h)$$

Obtenção do dominante

$$A\alpha\overline{A}\alpha\overline{\overline{A}}\alpha\underline{A} = 0 \quad (2.5i)$$
$$A\beta\overline{A}\beta\overline{\overline{A}}\beta\underline{A} = 1 \quad (2.5j)$$
$$A\gamma\overline{A}\gamma\overline{\overline{A}}\gamma\underline{A} = 2 \quad (2.5k)$$
$$A\delta\overline{A}\delta\overline{\overline{A}}\delta\underline{A} = 3 \quad (2.5l)$$

Distributividade crescente

$$(A\alpha B)\beta(A\alpha C) = A\alpha(B\beta C) \quad (2.5m)$$
$$(A\beta B)\gamma(A\beta C) = A\beta(B\gamma C) \quad (2.5n)$$
$$(A\gamma B)\delta(A\gamma C) = A\gamma(B\delta C) \quad (2.5o)$$
$$(A\delta B)\alpha(A\delta C) = A\delta(B\alpha C) \quad (2.5p)$$

As outras opções de distributividade não são válidas.

A propriedade da absorção continua sendo válida.

O valor dominante pode ser considerado como “*verdadeiro*” e o valor secundário como “*falso*”, ou seja, a negação do valor dominante, pois do deslocamento TOPO é a negação cíclica de Post.

A palavra “*função binária*”, numa definição específica para este contexto, refere-se a qualquer função que forneça apenas dois valores vizinhos para qualquer combinação de entrada. Os dois valores devem ser consecutivos, de modo que, se um deles for considerado como dominante (verdadeiro), o outro será o secundário (falso), ou seja, aquele valor modificado por um deslocamento TOPO. Por exemplo, temos $(0,1)_2$, $(0,1)_3$, $(1,2)_3$, $(2,0)_3$, $(0,1)_4$, $(1,2)_4$, $(2,3)_4$, $(3,0)_4$, etc, o primeiro número é o dominante, o segundo é o secundário e o sufixo indica a base. No caso de $(0,2)_4$, por exemplo, não se trata de uma função binária, pois a negação do valor “0” é “1” e não “2”.

Da mesma forma, a palavra “*ternário*” pode ser aplicada a $(0,1,2)_3$ $(0,1,2)_4$, $(1,2,3)_4$, $(2,3,0)_4$, $(3,0,1)_4$, etc. No caso de $(0,1,3)_5$, por exemplo, não se trata de uma função ternária.

A conversão de uma função MVL para binário consiste em qualquer substituição de valores de forma que a função satisfaça à definição de função binária. As tabelas para algumas operações que fazem conversão para binário e ternário (dentre elas, as fórmulas 2.4a-c encontram-se no apêndice 1.

Capítulo 3
Síntese de funções MVL

3.1 Introdução

Toda função lógica (MVL ou booleana) pode ser definida de duas maneiras:

1. Tabela verdade
2. Expressão algébrica

A definição por tabela verdade permite uma compreensão rápida das características da função, dando uma noção intuitiva de seu comportamento. A minimização da função por meio da tabela verdade é imediata, ou seja, não são necessárias grandes manipulações para perceber a presença de linhas ou colunas completas, que corresponde a uma das técnicas de minimização utilizadas pelo aplicativo, como será explicado mais adiante. A desvantagem deste modo de exibição é que a maneira como os valores são obtidos por meio das combinações de entrada não é perceptível, impossibilitando a implementação em circuitos elétricos ou em qualquer outra forma de dispositivo.

A definição por expressão algébrica permite a verificação das leis de formação empregadas na formação da função; também permite escrever a função em uma linha, como sendo uma fórmula. Esta forma de exibição é útil quando se pretende implementar a função em um dispositivo físico (circuito integrado, por exemplo), pois, a partir das operações fundamentais, pode-se obter todas as possibilidades de funções. A desvantagem deste modo de exibição consiste na dificuldade de percepção do comportamento da função e na minimização não intuitiva (não visual), que requer o emprego de diversas propriedades lógicas.

Ao processo de conversão da visualização por tabela verdade para a visualização por expressão algébrica dá-se o nome de síntese. A síntese pode ou não utilizar técnicas de minimização.

A lógica MVL ainda não foi totalmente formalizada, ainda há muitas características e propriedades a serem descobertas sobre esta parte da matemática. Diversas análises algébricas são impossíveis de se realizar utilizando-se apenas as leis verificadas até o momento. Algumas demonstrações e transformações podem ser feitas somente por meio da aferição por tabela-verdade.

A síntese de funções MVL a partir da tabela-verdade é feita de maneira parecida com a técnica para dois valores. Entretanto, ainda não se verificou uma utilidade para um código *Gray* no processo de síntese de funções MVL muito embora a idéia de se agrupar elementos da tabela formando retângulos, tal como é feito no mapa de Karnaugh para lógica booleana, já tenha sido empregada em outras técnicas de síntese de funções MVL [12]. Algumas propostas para códigos *Gray* MVL encontram-se no apêndice 1.

Não se dispõe, até o momento, de uma técnica de síntese de funções MVL pela lógica de Post estendida na qual se trabalhe diretamente com multi-valores, o que seria o ideal.

Uma técnica de síntese também proposta por Martins consiste em converter a função MVL em várias funções binárias [09] que, operadas adequadamente, fornecem a função original. Sendo assim, a síntese é feita sobre estas sub-funções binárias e não na função MVL. Estas sub-funções, apesar de serem binárias (usam apenas dois valores consecutivos), não aceitam o tratamento por mapa de Karnaugh, pois as variáveis pertencem ao domínio MVL.

Para uma base de *n* valores, tem-se *n* possibilidades de sub-funções binárias.

Para quatro valores, existe a possibilidade de se utilizar sub-função ternária, no entanto, tal sub-função deve ser sub-dividida em sub-funções binárias.

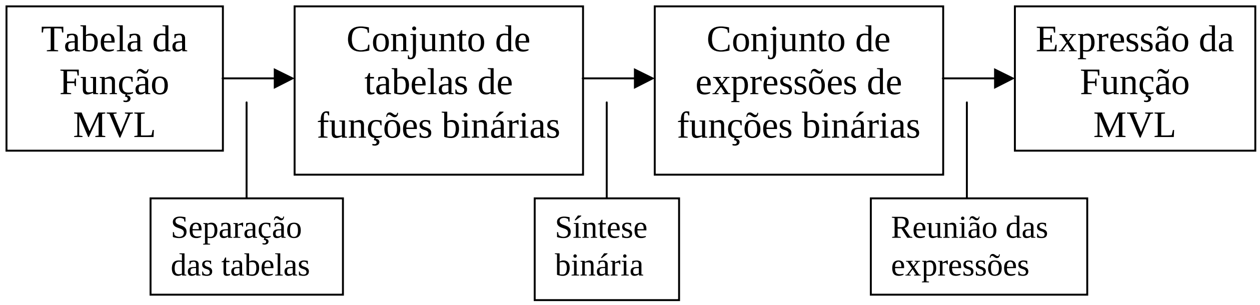


Figura 3.1

A simplificação feita na tabela das sub-funções é a de linhas ou colunas completas. Este tipo de simplificação pode ser feito tanto a partir da tabela-verdade como a partir da expressão algébrica.

Será verificado que determinadas ocorrências de alguns valores lógicos em certas sub-funções podem ser considerados como sendo irrelevantes (X), o que possibilita algumas simplificações por meio de substituições convenientes de tais ocorrências por outros valores.

Esta técnica, baseada na análise binária, apesar de ter tornado possível a síntese de funções MVL, gera alguns problemas. Por causa da conversão para funções binárias, a expressão final será formada por sub-funções binárias, e isso limita bastante as possibilidades para a escrita da expressão e nem sempre a forma mínima consiste em duas ou mais sub-funções binárias (quase nunca é). Por esse motivo, este método não permite a obtenção segura da forma mínima geral. Quando se falar em forma mínima nos textos a seguir, referir-se-á à forma mínima obtida por esta técnica.

O surgimento de valores irrelevantes nas sub-funções binárias é a causa do aumento da complexidade da função. Qualquer expressão algébrica pode ser ampliada adicionando-se redundâncias. Toda redundância aplicada aumenta o tamanho da expressão. Um método cem por cento minimizador não pode criar nenhum valor irrelevante. Uma vez que irrelevâncias aparecem para qualquer exemplo de função que utilize mais do que dois valores, conclui-se que esta técnica não fornece expressões cem por cento minimizadas para essas funções.

Por exemplo, a função *A*, para três valores (*buffer*), obtida por meio da matriz (012), que deveria ser a própria função *A*, é fornecida nas formas abaixo, cujo processo de obtenção será visto mais adiante.

$$(A\alpha 1)\gamma\left(A\beta\overline{A}\right) = (011)\ \gamma(112) = 012$$

$$(A\beta 2)\alpha\left(A\gamma\overline{A}\right) = (212)\ \alpha(022) = 012$$

$$(A\gamma 0)\beta\left(A\alpha\overline{A}\right) = (002)\ \beta(010) = 012$$

Realmente, o método funciona, pois as três funções correspondem à função fornecida, mas não foram expressas na forma mínima, que é “*A*”. Apesar de não ser o ideal, esta técnica é a única que existe no momento para este sistema lógico formal e sua criação proporcionou muitas facilidades no estudo da lógica multi-valores, bem como a implementação de circuitos MVL.

Por padronização, as funções MVL são denotadas por:

- Uma entrada: $B=B(A)$ (3.1a)
- Duas entradas: $C=C(A,B)$ (3.1b)
- Três entradas: $D=D(A,B,C)$ (3.1c)

Foi feita a escolha de se utilizar as colunas (eixo *X*) para a variável *A* e as linhas (eixo *Y*) para a variável *B* (quando houver). Sendo assim, temos as simplificações para linhas e colunas completas para duas entradas:

- $C_{COLUNA} = C(A)$ (3.2a)
- $C_{LINHA} = C(B)$ (3.2b)

Para funções de um argumento, a tabela-verdade corresponde a um vetor unidimensional; para dois argumentos, temos um vetor bidimensional, e assim por diante, sendo a dimensão do vetor a quantidade de argumentos.

Na lógica de Boole e na de Post, a função é constituída por vários termos. A palavra "*termo*" refere-se a uma combinação das variáveis de entrada de uma função binária (ver definição de função binária na página 26) para a qual a saída corresponde ao valor dominante para a operação utilizada entre os termos (também chamado de valor verdadeiro ou valor analisado). Para a lógica booleana, tem-se:

- Lógica negativa – analisa-se os zeros – forma disjuntiva - maxtermos
- Lógica positiva – analisa-se os uns – forma conjuntiva – mintermos

A palavra "maxtermo" indica tratar-se de um termo constituído pelo operador "máximo" – "$\vee$" que, para lógica binária, consiste no conectivo "*or*" – "+". A palavra "mintermo" indica tratar-se de um termo constituído pelo operador "mínimo" – "$\wedge$" que, para lógica binária, consiste no conectivo "*and*" – "*".

A razão pela qual esses conectivos são utilizados é aplicada também para a lógica MVL e é explicada a seguir.

- **Unicidade**: Cada termo (ocorrência do valor dominante ou verdadeiro) da tabela binária deve corresponder a uma expressão que seja verdadeira (valor dominante) se e somente se o valor das variáveis (coordenadas) corresponder à posição do termo na tabela (matriz). Para que isso aconteça, o conectivo utilizado entre as coordenadas deve ser aquele para o qual o valor analisado seja o indiferente, pois, dessa maneira, o termo somente será verdadeiro para uma única combinação de valores de entrada, aquela que corresponde a todas as entradas iguais ao valor indiferente.
- **Existência**: A expressão binária completa, formada pela junção das expressões relativas a cada termo, deve ser verdadeira se e somente se o valor das variáveis corresponder às coordenadas de um de seus termos. Para que isso aconteça, o conectivo utilizado entre os termos deve ser aquele para o qual o valor analisado seja o dominante, pois, dessa maneira, toda vez que um dos termos for verdadeiro, a expressão será verdadeira.

Toda expressão binária utiliza dois conectivos ou operadores, que são:

- Operação entre coordenadas
- Operação entre termos

Funções de uma entrada utilizam apenas a operação entre termos, pois os termos possuem apenas uma variável, não possuindo operação entre coordenadas.

A operação entre coordenadas é determinada pelo conceito da unicidade, e a operação entre termos, pelo conceito da existência.

3.2 Síntese de funções de dois valores e duas entradas

As funções equivalentes a linhas e colunas completas podem ser obtidas diretamente das tabelas abaixo, provenientes dos termos das funções de uma entrada, pois, em se tratando de uma linha ou coluna completa, deduz-se que uma das variáveis é irrelevante, temos uma função de apenas uma entrada.

COLUNAS COMPLETAS

C_1 - Zeros

0	1
A	$\overline{A}$

C_2 - Uns

0	1
$\overline{A}$	A

LINHAS COMPLETAS

C_1 - Zeros

0	1
B	$\overline{B}$

C_2 - Uns

0	1
$\overline{B}$	B

Tabela 3.1 – Expressões para colunas e linhas completas.

Um termo isolado, para duas entradas, faz parte de uma matriz bidimensional e, por isso, deve ser expresso em termos de suas coordenadas (*x*,*y*), que, no caso, são chamadas de (*A*,*B*). O valor das coordenadas é expresso em termos de deslocamentos.

A escolha dos operadores a serem utilizados, para quaisquer quantidades de valores, deve levar em consideração a idéia da unicidade e existência:

Forma	Sub-função	Operação entre coordenadas	Operação entre termos
C_1	*c0*	β	α
C_2	*c1*	α	β

Tabela 3.2

Comparando com a lógica booleana, tem-se:

Forma	Verdadeiro	Operação entre coordenadas	Operação entre termos
Disjuntiva	*0*	*Or (+)*	*And (*)*
Conjuntiva	*1*	*And (*)*	*Or (+)*

Tabela 3.3 – Comparação binário/dois valores.

Termos isolados podem ser obtidos a partir da tabela abaixo.

c0 - Zeros

0	*A = 0*	*A = 1*
B = 0	$A\beta B$	$\overline{A}\beta B$
B = 1	$A\beta\overline{B}$	$\overline{A}\beta\overline{B}$

c1 – Uns

1	*A = 0*	*A = 1*
B = 0	$\overline{A}\gamma\overline{B}$	$A\gamma\overline{B}$
B = 1	$\overline{A}\gamma B$	$A\gamma B$

Tabela 3.4 – Termos isolados.

A quantidade de opções para expressar a função depende da quantidade de valores que ela fornece, como é mostrado abaixo:

Valores	C_1	C_2
0	Sim	
1		Sim
0,1	Sim	Sim

Tabela 3.5

Exemplo:

C \ A	**0**	**1**
B **0**	0	0
1	0	1

Tabela 3.6

$$C_1 = (A\beta B)\alpha(\overline{A}\beta B)\alpha(A\beta\overline{B})$$
$$C_2 = (A\alpha B)$$

São, ao todo, $2^4 = 16$ funções de três valores e duas entradas. As funções de dois valores foram incluídas neste estudo apenas para mostrar que a técnica utilizada na síntese de funções MVL é uma expansão daquela utilizada na lógica binária.

3.3 Síntese de funções de três valores e uma entrada

Toda função de três valores pode ser decomposta em três sub-funções binárias. No caso de função de uma entrada $B=B(A)$, as sub-funções são $b0$, $b1$, $b2$. Cada uma dessas funções fornece apenas dois valores, sendo um deles o valor a ser analisado (dominante), valor para o qual cada ocorrência fornece um termo, exatamente como é feito na lógica binária nas formas disjuntiva e conjuntiva.

Sub-função binária	Dominante Verdadeiro	Secundário Falso	Operação entre termos
b0	0	1	α
b1	1	2	β
b2	2	0	γ

Tabela 3.7 – Operações entre termos.

Se a função analisada fornecer apenas dois valores, ela será representada diretamente por uma das três sub-funções $b0$, $b1$, $b2$, não sendo necessária a análise das outras duas sub-funções.

A função $B=B(A)$ pode ser obtida de três formas:

$B_1=B_1(A)$ (3.3*a*) $B_1=B_1(b0,b1)$ (3.3*d*) $b0=b0(A)$ (3.3*g*)

$B_2=B_2(A)$ (3.3*b*) $B_2=B_2(b1,b2)$ (3.3*e*) $b1=b1(A)$ (3.3*h*)

$B_3=B_3(A)$ (3.3*c*) $B_3=B_3(b2,b0)$ (3.3*f*) $b2=b2(A)$ (3.3*i*)

Para converter a função $B=B(A)$ nas sub-funções $b0,b1,b2$, faz-se a substituição do valor indiferente pelo valor dominante.

Sub-função	Substituição
b0	$2 \rightarrow 0$
b1	$0 \rightarrow 1$
b2	$1 \rightarrow 2$

Tabela 3.8

	b0	*b1*	*b2*
0	0	$1_{(0)}$	0
1	1	1	$2_{(1)}$
2	$0_{(2)}$	2	2

Tabela 3.9

	b0	*b1*	*b2*
Dominante	0	1	2
Secundário	1	2	0
Indiferente	$0_{(2)}$	$1_{(0)}$	$2_{(1)}$

Tabela 3.10

Lembrando dos operadores para a realização da conversão para binário ou eliminação do valor indiferente: ($A\alpha 1$, $A\beta 2$, $A\gamma 0$), as expressões equivalentes aos termos podem ser obtidas diretamente das tabelas abaixo:

b0- Zeros

0	1	2
$A\alpha 1$	$\underline{A}\alpha 1$	$\overline{A}\alpha 1$

b1- Uns

0	1	2
$\overline{A}\beta 2$	$A\beta 2$	$\underline{A}\beta 2$

b2- Dois

0	1	2
$\underline{A}\gamma 0$	$\overline{A}\gamma 0$	$A\gamma 0$

Tabela 3.11 – Termos.

Quando houver dois termos ou mais, não é necessário fazer a conversão para binário, como mostra a tabela ao lado:

Termos	dominante	Secundário	indiferente	c.p.b.
1	1	1	1	Sim
2	2	1		Não
3	3			Não

Tabela 3.12 – Necessidade de fazer a conversão para binário.

A função completa utiliza, entre as sub-funções binárias, o conectivo para a qual os valores analisados nas duas sub-funções sejam o secundário e o indiferente, respectivamente. Outro raciocínio consiste em considerar as duas sub-funções binárias como sendo termos para a função completa; sendo assim, o conectivo (entre termos) a ser utilizado é aquele que ainda não foi utilizado entre os termos das duas sub-funções.

Forma	Sub-funções binárias	Analisam, respectivamente	Operações entre termos das sub-funções	Operação entre sub-funções	Dominante do operador
B_1	($b0$, $b1$)	(0,1)	(α, β)	γ	2
B_2	($b1$, $b2$)	(1,2)	(β, γ)	α	0
B_3	($b2$, $b0$)	(2,0)	(γ, α)	β	1

Tabela 3.13 – Operações entre sub-funções.

Sendo assim, as três formas da função são definidas como:

$$B_1 = b0\ \gamma\ b1 \quad (3.4a)$$

$$B_2 = b1\ \alpha\ b2 \quad (3.4b)$$

$$B_3 = b2\ \beta\ b0 \quad (3.4c)$$

	1ª	2ª
B_1	$b0$	$b1$
0	0	$1_{(0)}$
1	1	1
2	$0_{(2)}$	2

	1ª	2ª
B_2	$b1$	$b2$
0	$1_{(0)}$	0
1	1	$2_{(1)}$
2	2	2

	1ª	2ª
B_3	$b2$	$b0$
0	0	0
1	$2_{(1)}$	1
2	2	$0_{(2)}$

Tabela 3.14

Levando-se em conta as equações 3.4, percebe-se que o valor indiferente na primeira sub-função é operado com o valor dominante (da operação utilizada entre as duas sub-funções) na segunda sub-função. Por esse motivo, o valor indiferente na primeira sub-função torna-se irrelevante, podendo ser substituído por qualquer um dos três valores. Obviamente, o valor irrelevante não pode ser substituído por ele mesmo, pois, nesse caso, se obtém uma função binária. Pode-se fazer a substituição pelo dominante ou pelo secundário, mas, como cada ocorrência do dominante acarreta em um termo na função, deve-se fazer a substituição pelo secundário.

Por convenção, quando a sub-função é obtida por meio da substituição por um irrelevante, ela é chamada de b^*.

	1ª	2ª
$B_1(\gamma)$	*b0*	*b1*
2	$0_{(2)}$	2

$X\gamma 2 = 2$

	1ª	2ª
$B_2(\alpha)$	*b1*	*b2*
0	$1_{(0)}$	0

$X\alpha 0 = 0$

	1ª	2ª
$B_3(\beta)$	*b2*	*b0*
1	$2_{(1)}$	1

$X\beta 1 = 1$

Tabela 3.15

	1ª	2ª
B_1	$b0^*$	*b1*
0	0	$1_{(0)}$
1	1	1
2	$1_{(2)}$	2

	1ª	2ª
B_2	$b1^*$	*b2*
0	$2_{(0)}$	0
1	1	$2_{(1)}$
2	2	2

	1ª	2ª
B_3	$b2^*$	*b0*
0	0	0
1	$0_{(1)}$	1
2	2	$0_{(2)}$

Tabela 3.16

1ª	$b0^*$	$b1^*$	$b2^*$
Dominante	0	1	2
Secundário	1	2	0
Indiferente	$1_{(2)}$	$2_{(0)}$	$0_{(1)}$

2ª	*b0*	*b1*	*b2*
Dominante	0	1	2
Secundário	1	2	0
Indiferente	$0_{(2)}$	$1_{(0)}$	$2_{(1)}$

Tabela 3.17

De uma maneira geral, tem-se (o número ordinal indica a hierarquia de dominância):

	1ª	2ª
B_1	b^*	*b*
0	1º	1º (3o)
1	2º	1º
2	2º (3o)	2º

	1ª	2ª
	b^*	*b*
Dominante	1º	1º
Secundário	2º	2º
Indiferente	2º (3o)	1º (3o)

Tabela 3.18

1ª	b^*
Dominante	Dominante
Secundário	Secundário
Indiferente	Secundário

2ª	b
Dominante	Dominante
Secundário	Secundário
Indiferente	Dominante

Tabela 3.19

$$B_1 = b0^* \, \gamma \, b1 \quad (3.5a)$$

$$B_2 = b1^* \, \alpha \, b2 \quad (3.5b)$$

$$B_3 = b2^* \, \beta \, b0 \quad (3.5c)$$

A quantidade de sub-funções necessárias para expressar a função depende da quantidade de valores que ela fornece, como é mostrado abaixo:

Valores	*b0*	*b1*	*b2*	Sub-funções
0 ou (0,1)	Sim			1
1 ou (1,2)		Sim		1
2 ou (2,0)			Sim	1
0,1,2	Sim	Sim	Sim	2

Tabela 3.20 – Sub-funções necessárias.

Será mostrada, agora, a análise das $3^3 = 27$ funções de três valores e uma entrada por ordem de simplicidade.

Uma vez que são utilizadas somente sub-funções binárias, pode-se tabelar tais funções e utilizá-las na criação da função completa. São $2^3 = 8$ sub-funções binárias. Se se considerar as três variações (0,1) (*b0*), (1,2) (*b1*) e (2,0) (*b2*); tem-se 24 sub-funções.

B	$b0$	B	$b0$
000	0	100	$\overline{A}\alpha A$
001	$\underline{A}\alpha A$	101	$\underline{A}\alpha 1$
010	$A\alpha\overline{A}$	110	$\overline{A}\alpha 1$
011	$A\alpha 1$	111	1

Tabela 3.21 – Exemplo de *b0*.

As funções mostradas na tabela abaixo fornecem apenas o valor dominante e o secundário do conectivo utilizado. Estas funções correspondem a funções binárias e, por isso, requerem o uso de apenas uma sub-função. Estas são as 18 funções básicas e são utilizadas na síntese de funções que forneçam os três valores.

B	$b0$	B	$b1$	B	$b2$
011	$A\alpha 1$	212	$A\beta 2$	002	$A\gamma 0$
110	$\overline{A}\alpha 1$	122	$\overline{A}\beta 2$	020	$\overline{A}\gamma 0$
101	$\underline{A}\alpha 1$	221	$\underline{A}\beta 2$	200	$\underline{A}\gamma 0$
010	$A\alpha\overline{A}$	112	$A\beta\overline{A}$	022	$A\gamma\overline{A}$
100	$\overline{A}\alpha\underline{A}$	121	$\overline{A}\beta\underline{A}$	220	$\overline{A}\gamma\underline{A}$
001	$\underline{A}\alpha A$	211	$\underline{A}\beta A$	202	$\underline{A}\gamma A$

Tabela 3.22 – As 18 funções básicas.

Estas funções podem ser nomeadas como mostrado abaixo:

f_0	$A\alpha 1$ 011	f_3	$A\beta 2$ 212	f_6	$A\gamma 0$ 002	g_0	$A\alpha\overline{A}$ 010	g_3	$A\beta\overline{A}$ 112	g_6	$A\gamma\overline{A}$ 022
f_1	$\overline{A}\alpha 1$ 110	f_4	$\overline{A}\beta 2$ 122	f_7	$\overline{A}\gamma 0$ 020	g_1	$\overline{A}\alpha\underline{A}$ 100	g_4	$\overline{A}\beta\underline{A}$ 121	g_7	$\overline{A}\gamma\underline{A}$ 220
f_2	$\underline{A}\alpha 1$ 101	f_5	$\underline{A}\beta 2$ 221	f_8	$\underline{A}\gamma 0$ 200	g_2	$\underline{A}\alpha A$ 001	g_5	$\underline{A}\beta A$ 211	g_8	$\underline{A}\gamma A$ 202

Tabela 3.23 – Nomenclatura das 18 funções básicas.

As funções mostradas na tabela abaixo fornecem os três valores e, por isso, requerem o uso das três sub-funções. As funções binárias vistas anteriormente podem ser combinadas entre si, gerando estas funções.

Nas funções da tabela abaixo, optou-se por substituir o valor indiferente do primeiro termo (irrelevante) pelo valor dominante.

B	$b0$	$b1$	$b2$	$B_1 = b0\,\gamma\,b1$	$B_2 = b1\,\alpha\,b2$	$B_3 = b2\,\beta\,b0$
A 012	$A\alpha\overline{A}$ 010	$A\beta\overline{A}$ 112	$A\gamma\overline{A}$ 022	$(A\alpha\overline{A})\gamma(A\beta\overline{A})$ (010) γ(112)	$(A\beta\overline{A})\alpha(A\gamma\overline{A})$ (112) α(022)	$(A\gamma\overline{A})\beta(A\alpha\overline{A})$ (022) β(010)
$\overline{A}$ 120	$\overline{A}\alpha\underline{A}$ 100	$\overline{A}\beta\underline{A}$ 121	$\overline{A}\gamma\underline{A}$ 220	$(\overline{A}\alpha\underline{A})\gamma(\overline{A}\beta\underline{A})$ (100) γ(121)	$(\overline{A}\beta\underline{A})\alpha(\overline{A}\gamma\underline{A})$ (121) α(220)	$(\overline{A}\gamma\underline{A})\beta(\overline{A}\alpha\underline{A})$ (220) β(100)
$\underline{A}$ 201	$\underline{A}\alpha A$ 001	$\underline{A}\beta A$ 211	$\underline{A}\gamma A$ 202	$(\underline{A}\alpha A)\gamma(\underline{A}\beta A)$ (001) γ(211)	$(\underline{A}\beta A)\alpha(\underline{A}\gamma A)$ (211) α(202)	$(\underline{A}\gamma A)\beta(\underline{A}\alpha A)$ (202) β(001)
210	$A\alpha\overline{A}$ 010	$\underline{A}\beta A$ 211	$\overline{A}\gamma\underline{A}$ 220	$(A\alpha\overline{A})\gamma(\underline{A}\beta A)$ (010) γ(211)	$(\underline{A}\beta A)\alpha(\overline{A}\gamma\underline{A})$ (211) α(220)	$(\overline{A}\gamma\underline{A})\beta(A\alpha\overline{A})$ (220) β(010)
102	$\overline{A}\alpha\underline{A}$ 100	$A\beta\overline{A}$ 112	$\underline{A}\gamma A$ 202	$(\overline{A}\alpha\underline{A})\gamma(A\beta\overline{A})$ (100) γ(112)	$(A\beta\overline{A})\alpha(\underline{A}\gamma A)$ (112) α(202)	$(\underline{A}\gamma A)\beta(\overline{A}\alpha\underline{A})$ (202) β(100)
021	$\underline{A}\alpha A$ 001	$\overline{A}\beta\underline{A}$ 121	$A\gamma\overline{A}$ 022	$(\underline{A}\alpha A)\gamma(\overline{A}\beta\underline{A})$ (001) γ(121)	$(\overline{A}\beta\underline{A})\alpha(A\gamma\overline{A})$ (121) α(022)	$(A\gamma\overline{A})\beta(\underline{A}\alpha A)$ (022) β(001)

Tabela 3.24 – As 6 funções que fornecem 3 valores.

Lembrando que, no primeiro termo o valor indiferente pode ser substituído pelo secundário, pode-se fazer a seguinte análise:

B	$b0^*$	$b1$	$B_1 = b0^* \gamma b1$
022	$A\alpha 1$ 011	$\overline{A}\beta 2$ 122	$(A\alpha 1)\gamma(\overline{A}\beta 2)$
220	$\overline{A}\alpha 1$ 110	$\underline{A}\beta 2$ 221	$(\overline{A}\alpha 1)\gamma(\underline{A}\beta 2)$
202	$\underline{A}\alpha 1$ 101	$A\beta 2$ 212	$(\underline{A}\alpha 1)\gamma(A\beta 2)$
020	$A\alpha\overline{A}$ 010	$\overline{A}\beta\underline{A}$ 121	$(A\alpha\overline{A})\gamma(\overline{A}\beta\underline{A})$
200	$\overline{A}\alpha\underline{A}$ 100	$\underline{A}\beta A$ 211	$(\overline{A}\alpha\underline{A})\gamma(\underline{A}\beta A)$
002	$\underline{A}\alpha A$ 001	$A\beta\overline{A}$ 112	$(\underline{A}\alpha A)\gamma(A\beta\overline{A})$

B	$b1^*$	$b2$	$B_2 = b1^* \alpha b2$
010	$A\beta 2$ 212	$\overline{A}\gamma 0$ 020	$(A\beta 2)\alpha(\overline{A}\gamma 0)$
100	$\overline{A}\beta 2$ 122	$\underline{A}\gamma 0$ 200	$(\overline{A}\beta 2)\alpha(\underline{A}\gamma 0)$
001	$\underline{A}\beta 2$ 221	$A\gamma 0$ 002	$(\underline{A}\beta 2)\alpha(A\gamma 0)$
110	$A\beta\overline{A}$ 112	$\overline{A}\gamma\underline{A}$ 220	$(A\beta\overline{A})\alpha(\overline{A}\gamma\underline{A})$
101	$\overline{A}\beta\underline{A}$ 121	$\underline{A}\gamma A$ 202	$(\overline{A}\beta\underline{A})\alpha(\underline{A}\gamma A)$
011	$\underline{A}\beta A$ 211	$A\gamma\overline{A}$ 022	$(\underline{A}\beta A)\alpha(A\gamma\overline{A})$

B	$b2^*$	$b0$	$B_3 = b2^* \beta b0$
112	$A\gamma 0$ 002	$\overline{A}\alpha 1$ 110	$(A\gamma 0)\beta(\overline{A}\alpha 1)$
121	$\overline{A}\gamma 0$ 020	$\underline{A}\alpha 1$ 101	$(\overline{A}\gamma 0)\beta(\underline{A}\alpha 1)$
211	$\underline{A}\gamma 0$ 200	$A\alpha 1$ 011	$(\underline{A}\gamma 0)\beta(A\alpha 1)$
122	$A\gamma\overline{A}$ 022	$\overline{A}\alpha\underline{A}$ 100	$(A\gamma\overline{A})\beta(\overline{A}\alpha\underline{A})$
221	$\overline{A}\gamma\underline{A}$ 220	$\underline{A}\alpha A$ 001	$(\overline{A}\gamma\underline{A})\beta(\underline{A}\alpha A)$
212	$\underline{A}\gamma A$ 202	$A\alpha\overline{A}$ 010	$(\underline{A}\gamma A)\beta(A\alpha\overline{A})$

Tabela 3.25

B	$b0^*$	$b1$	$B_1 = b0^* \gamma b1$	$b1^*$	$b2$	$B_2 = b1^* \alpha b2$	$b2^*$	$b0$	$B_3 = b2^* \beta b0$
012	$A\alpha 1$ 011	$A\beta\overline{A}$ 112	$(A\alpha 1)\gamma(A\beta\overline{A})$	$A\beta 2$ 212	$A\gamma\overline{A}$ 022	$(A\beta 2)\alpha(A\gamma\overline{A})$	$A\gamma 0$ 002	$A\alpha\overline{A}$ 010	$(A\gamma 0)\beta(A\alpha\overline{A})$
120	$\overline{A}\alpha 1$ 110	$\overline{A}\beta\underline{A}$ 121	$(\overline{A}\alpha 1)\gamma(\overline{A}\beta\underline{A})$	$\overline{A}\beta 2$ 122	$\overline{A}\gamma\underline{A}$ 220	$(\overline{A}\beta 2)\alpha(\overline{A}\gamma\underline{A})$	$\overline{A}\gamma 0$ 020	$\overline{A}\alpha\underline{A}$ 100	$(\overline{A}\gamma 0)\beta(\overline{A}\alpha\underline{A})$
201	$\underline{A}\alpha 1$ 101	$\underline{A}\beta A$ 211	$(\underline{A}\alpha 1)\gamma(\underline{A}\beta A)$	$\underline{A}\beta 2$ 221	$\underline{A}\gamma A$ 202	$(\underline{A}\beta 2)\alpha(\underline{A}\gamma A)$	$\underline{A}\gamma 0$ 200	$\underline{A}\alpha A$ 001	$(\underline{A}\gamma 0)\beta(\underline{A}\alpha A)$
021	$A\alpha 1$ 011	$\overline{A}\beta\underline{A}$ 121	$(A\alpha 1)\gamma(\overline{A}\beta\underline{A})$	$\underline{A}\beta 2$ 221	$A\gamma\overline{A}$ 022	$(\underline{A}\beta 2)\alpha(A\gamma\overline{A})$	$\overline{A}\gamma 0$ 020	$\underline{A}\alpha A$ 001	$(\overline{A}\gamma 0)\beta(\underline{A}\alpha A)$
210	$\overline{A}\alpha 1$ 110	$\underline{A}\beta A$ 211	$(\overline{A}\alpha 1)\gamma(\underline{A}\beta A)$	$A\beta 2$ 212	$\overline{A}\gamma\underline{A}$ 220	$(A\beta 2)\alpha(\overline{A}\gamma\underline{A})$	$\underline{A}\gamma 0$ 200	$A\alpha\overline{A}$ 010	$(\underline{A}\gamma 0)\beta(A\alpha\overline{A})$
102	$\underline{A}\alpha 1$ 101	$A\beta\overline{A}$ 112	$(\underline{A}\alpha 1)\gamma(A\beta\overline{A})$	$\overline{A}\beta 2$ 122	$\underline{A}\gamma A$ 202	$(\overline{A}\beta 2)\alpha(\underline{A}\gamma A)$	$A\gamma 0$ 002	$\overline{A}\alpha\underline{A}$ 100	$(A\gamma 0)\beta(\overline{A}\alpha\underline{A})$

Tabela 3.26 – As 6 funções que fornecem 3 valores.

As funções ternárias que fornece todos os três valores consistem em (012) e (210) e seus deslocamentos. A primeira consiste na própria entrada, a segunda corresponde à negação de Lukasiewicz.

As funções podem ser construídas a partir das sub-funções g e f.

	A	*b0*	*b1*	*b2*	$B_1 = b0\ \gamma\ b1$	$B_2 = b1\ \alpha\ b2$	$B_3 = b2\ \beta\ b0$
	020	g_0	g_4		$B_1 = g_0\ \gamma\ g_4$		
	200	g_1	g_5		$B_1 = g_1\ \gamma\ g_5$		
	002	g_2	g_3		$B_1 = g_2\ \gamma\ g_3$		
	110		g_3	g_7		$B_2 = g_3\ \alpha\ g_7$	
Tabela 3.25	101		g_4	g_8		$B_2 = g_4\ \alpha\ g_8$	
	011		g_5	g_6		$B_2 = g_5\ \alpha\ g_6$	
	122	g_1		g_6			$B_3 = g_6\ \beta\ g_1$
	221	g_2		g_7			$B_3 = g_7\ \beta\ g_2$
	212	g_0		g_5			$B_3 = g_5\ \beta\ g_0$
	012	g_0	g_3	g_6	$B_1 = g_0\ \gamma\ g_3$	$B_2 = g_3\ \alpha\ g_6$	$B_3 = g_6\ \beta\ g_0$
	120	g_1	g_4	g_7	$B_1 = g_1\ \gamma\ g_4$	$B_2 = g_4\ \alpha\ g_7$	$B_3 = g_7\ \beta\ g_1$
	201	g_2	g_5	g_8	$B_1 = g_2\ \gamma\ g_5$	$B_2 = g_5\ \alpha\ g_8$	$B_3 = g_8\ \beta\ g_2$
Tabela 3.24	210	g_0	g_5	g_7	$B_1 = g_0\ \gamma\ g_5$	$B_2 = g_5\ \alpha\ g_7$	$B_3 = g_7\ \beta\ g_0$
	102	g_1	g_3	g_8	$B_1 = g_1\ \gamma\ g_3$	$B_2 = g_3\ \alpha\ g_8$	$B_3 = g_8\ \beta\ g_1$
	021	g_2	g_4	g_6	$B_1 = g_2\ \gamma\ g_4$	$B_2 = g_4\ \alpha\ g_6$	$B_3 = g_6\ \beta\ g_2$

Tabela 3.27

Como pode ser visto, por meio de apenas nove funções binárias ($g_{0\text{-}8}$), obtém-se todas as 27 funções de três valores e uma entrada.

	A	*b0*	*b1*	*b2*	$B_1 = b0\,\gamma\,b1$	$B_2 = b1\,\alpha\,b2$	$B_3 = b2\,\beta\,b0$
	022	f_0	f_4		$B_1 = f_0\,\gamma\,f_4$		
	220	f_1	f_5		$B_1 = f_1\,\gamma\,f_5$		
	202	f_2	f_3		$B_1 = f_2\,\gamma\,f_3$		
	010		f_3	f_7		$B_2 = f_3\,\alpha\,f_7$	
Tabela 3.25	100		f_4	f_8		$B_2 = f_4\,\alpha\,f_8$	
	001		f_5	f_6		$B_2 = f_5\,\alpha\,f_6$	
	112	f_1		f_6			$B_3 = f_6\,\beta\,f_1$
	121	f_2		f_7			$B_3 = f_7\,\beta\,f_2$
	211	f_0		f_8			$B_3 = f_8\,\beta\,f_0$
	012	f_0	f_3	f_6	$B_1 = f_0\,\gamma\,g_3$	$B_2 = f_3\,\alpha\,g_6$	$B_3 = f_6\,\beta\,g_0$
	120	f_1	f_4	f_7	$B_1 = f_1\,\gamma\,g_4$	$B_2 = f_4\,\alpha\,g_7$	$B_3 = f_7\,\beta\,g_1$
Tabela 3.26	201	f_2	f_5	f_8	$B_1 = f_2\,\gamma\,g_5$	$B_2 = f_5\,\alpha\,g_8$	$B_3 = f_8\,\beta\,g_2$
	210	f_0	f_5	f_7	$B_1 = f_0\,\gamma\,g_4$	$B_2 = f_5\,\alpha\,g_6$	$B_3 = f_7\,\beta\,g_2$
	102	f_1	f_3	f_8	$B_1 = f_1\,\gamma\,g_5$	$B_2 = f_3\,\alpha\,g_7$	$B_3 = f_8\,\beta\,g_0$
	021	f_2	f_4	f_6	$B_1 = f_2\,\gamma\,g_3$	$B_2 = f_4\,\alpha\,g_8$	$B_3 = f_6\,\beta\,g_1$

Tabela 3.28

3.4 Síntese de funções de três valores e duas entradas

Todas as regras para síntese de funções de três valores e uma entrada são aplicáveis à síntese de funções de três valores e duas entradas. A função é $C=C(A,B)$, as sub-funções são $c0,c1,c2$. Se a função analisada fornecer apenas dois valores, ela será representada diretamente por uma das três sub-funções $c0,c1,c2$, não sendo necessária a análise das outras duas sub-funções.

A função $C=C(A,B)$ pode ser obtida de três formas:

$C_1=C_1(A,B)$ (3.6*a*) $C_1=C_1(c0,c1)$ (3.6*d*) $c0=c0(A,B)$ (3.6*g*) $C_1 = c0\ \gamma\ c1$ (3.6*j*)

$C_2=C_2(A,B)$ (3.6*b*) $C_2=C_2(c1,c2)$ (3.6*e*) $c1=c1(A,B)$ (3.6*h*) $C_2 = c1\ \alpha\ c2$ (3.6*k*)

$C_3=C_3(A,B)$ (3.6*c*) $C_3=C_3(c2,c0)$ (3.6*f*) $c2=c2(A,B)$ (3.6*i*) $C_3 = c2\ \beta\ c0$ (3.6*l*)

Para converter a função $C=C(A,B)$ nas sub-funções $c0,c1,c2$, faz-se a substituição do valor indiferente pelo valor dominante.

Sub-função	Substituição
c0	$2 \rightarrow 0$
c1	$0 \rightarrow 1$
c2	$1 \rightarrow 2$

Tabela 3.29

	c0	*c1*	*c2*
0	0	$1_{(0)}$	0
1	1	1	$2_{(1)}$
2	$0_{(2)}$	2	2

Tabela 3.30

	c0	*c1*	*c2*
Dominante	0	1	2
Secundário	1	2	0
Indiferente	$0_{(2)}$	$1_{(0)}$	$2_{(1)}$

Tabela 3.31

As funções equivalentes a linhas e colunas completas podem ser obtidas diretamente das tabelas a seguir, provenientes dos termos das funções de uma entrada, pois, em se tratando de uma linha ou coluna completa, deduz-se que uma das variáveis seja irrelevante, ou seja, tem-se uma função de apenas uma entrada.

Abaixo, tem-se as expressões para colunas completas e linhas completas, respectivamente.

c0 - Zeros

0	1	2
$A\alpha 1$	$\underline{A}\alpha 1$	$\overline{A}\alpha 1$

c1 - Uns

0	1	2
$\overline{A}\beta 2$	$A\beta 2$	$\underline{A}\beta 2$

c2 - Dois

0	1	2
$\underline{A}\gamma 0$	$\overline{A}\gamma 0$	$A\gamma 0$

c0 - Zeros

0	1	2
$B\alpha 1$	$\underline{B}\alpha 1$	$\overline{B}\alpha 1$

c1 - Uns

0	1	2
$\overline{B}\beta 2$	$B\beta 2$	$\underline{B}\beta 2$

c2 - Dois

0	1	2
$\underline{B}\gamma 0$	$\overline{B}\gamma 0$	$B\gamma 0$

Tabela 3.32 – Termos equivalentes a colunas e linhas completas.

As demonstrações das funções de coluna completa são mostradas abaixo. As funções de linha completa podem ser demonstradas de maneira análoga.

$$(A\beta B)\alpha(A\beta\overline{B})\alpha(A\beta\underline{B}) = (A\alpha B)\beta(A\alpha\overline{B})\beta(A\alpha\underline{B}) = A\alpha(B\beta\overline{B}\beta\underline{B}) = A\alpha 1$$
$$(A\gamma B)\beta(A\gamma\overline{B})\beta(A\gamma\underline{B}) = (A\beta B)\gamma(A\beta\overline{B})\gamma(A\beta\underline{B}) = A\beta(B\gamma\overline{B}\gamma\underline{B}) = A\beta 2$$
$$(A\alpha B)\gamma(A\alpha\overline{B})\gamma(A\alpha\underline{B}) = (A\gamma B)\alpha(A\gamma\overline{B})\alpha(A\gamma\underline{B}) = A\gamma(B\alpha\overline{B}\alpha\underline{B}) = A\gamma 0$$

Um termo isolado, para duas entradas, faz parte de uma matriz bidimensional e, por isso, deve ser expresso em termos de suas coordenadas (*x*,*y*), que, no caso, são chamadas de (*A*,*B*). O valor das coordenadas é expresso em termos de deslocamentos.

Sub-função binária	Dominante Verdadeiro	Secundário Falso	Operação entre coordenadas	Operação entre termos
c0	0	1	β	α
c1	1	2	γ	β
c2	2	0	α	γ

Tabela 3.33 – Operações entre coordenadas e entre termos.

Os conectivos a serem utilizados entre as sub-funções são determinados pelos mesmos critérios empregados na síntese de funções de uma entrada.

Forma	Sub-funções binárias	Analisam, respectivamente	Operações entre termos das sub-funções	Operação entre sub-funções	Dominante do operador
C_1	$(c0, c1)$	(0,1)	(α, β)	γ	2
C_2	$(c1, c2)$	(1,2)	(β, γ)	α	0
C_3	$(c2, c0)$	(2,0)	(γ, α)	β	1

Tabela 3.34 – Operações entre sub-funções.

Sendo assim, é possível resumir o uso das operações usadas por cada sub-função e função. O símbolo “ ”, utilizado na tabela abaixo, representa qualquer variável de entrada deslocada de qualquer valor $\left(A, \overline{A}, \underline{A}, B, \overline{B}, \underline{B}\right)$.

	Operação entre coordenadas	Operação entre termos	Aspecto:
c0	$\beta(1)$	$\alpha(0)$	$(\quad\beta\quad)\alpha(\quad\beta\quad)$
c1	$\gamma(2)$	$\beta(1)$	$(\quad\gamma\quad)\beta(\quad\gamma\quad)$
c2	$\alpha(0)$	$\gamma(2)$	$(\quad\alpha\quad)\gamma(\quad\alpha\quad)$
C1	-	$\gamma(2)$	$[(\quad\beta\quad)\alpha(\quad\beta\quad)]\,\gamma\,[(\quad\gamma\quad)\beta(\quad\gamma\quad)]$
C2	-	$\alpha(0)$	$[(\quad\gamma\quad)\beta(\quad\gamma\quad)]\,\alpha\,[(\quad\alpha\quad)\gamma(\quad\alpha\quad)]$
C3	-	$\beta(1)$	$[(\quad\alpha\quad)\gamma(\quad\alpha\quad)]\,\beta\,[(\quad\beta\quad)\alpha(\quad\beta\quad)]$

Tabela 3.35

Termos isolados podem ser obtidos a partir da tabela mostrada a seguir. Para não sobrecarregar a tabela, a conversão para binário foi omitida; no entanto, ela sempre deve ser levada em consideração ($A\alpha 1$, $A\beta 2$, $A\gamma 0$).

c0 - Zeros

0	*A = 0*	*A = 1*	*A = 2*
B = 0	$A\beta B$	$\underline{A}\beta B$	$\overline{A}\beta B$
B = 1	$A\beta \underline{B}$	$\underline{A}\beta \underline{B}$	$\overline{A}\beta \underline{B}$
B = 2	$A\beta \overline{B}$	$\underline{A}\beta \overline{B}$	$\overline{A}\beta \overline{B}$

c1 – Uns

1	*A = 0*	*A = 1*	*A = 2*
B = 0	$\overline{A}\gamma \overline{B}$	$A\gamma \overline{B}$	$\underline{A}\gamma \overline{B}$
B = 1	$\overline{A}\gamma B$	$A\gamma B$	$\underline{A}\gamma B$
B = 2	$\overline{A}\gamma \underline{B}$	$A\gamma \underline{B}$	$\underline{A}\gamma \underline{B}$

c2 – Dois

2	*A = 0*	*A = 1*	*A = 2*
B = 0	$\underline{A}\alpha \underline{B}$	$\overline{A}\alpha \underline{B}$	$A\alpha \underline{B}$
B = 1	$\underline{A}\alpha \overline{B}$	$\overline{A}\alpha \overline{B}$	$A\alpha \overline{B}$
B = 2	$\underline{A}\alpha B$	$\overline{A}\alpha B$	$A\alpha B$

Tabela 3.36 – Termos.

Sempre que apenas uma das três linhas ou colunas contiver o valor analisado a conversão para binário será necessária, mesmo em se tratando de linha ou coluna completa (caso unidimensional).

Exemplo 1: Conectando-se os termos da tabela temos que $C = (A\beta B)\alpha(\underline{A}\beta \underline{B})$. Ao verificar esta função, percebe-se que ela realmente corresponde à função dada.

Fornecido

B \ A			
	0	1	1
B	1	**0**	1
	1	1	1

$A\beta B$

B \ A			
	0	1	2
B	1	1	1
	2	1	2

$\underline{A}\beta \underline{B}$

B \ A			
	2	2	1
B	2	**0**	1
	1	1	1

$(A\beta B)\alpha(\underline{A}\beta \underline{B})$

B \ A			
	0	1	1
B	1	**0**	1
	1	1	1

Tabela 3.37

Conclusão: Sempre que a função contiver pelo menos dois termos em linhas e colunas diferentes, em diagonal, a conversão para binário é dispensável.

Exemplo 2:

Fornecido

B \ A			
	0	**0**	1
B	1	1	1
	1	1	1

$A\beta B$

B \ A			
	0	1	2
B	1	1	1
	2	1	2

$\underline{A}\beta B$

B \ A			
	2	**0**	1
B	1	1	1
	2	2	1

$(A\beta B)\alpha(\underline{A}\beta B)$

B \ A			
	0	**0**	1
B	1	1	1
	2	1	1

Tabela 3.38

Conclusão: A recíproca é verdadeira, ou seja, se a condição mostrada no exemplo 1 não for satisfeita, a conversão para binário é necessária. Esta conclusão pode ser verificada analisando-se todos os casos.

A simplificação da primeira sub-função é a mesma vista para função de uma entrada (substituição pelo secundário). No entanto, uma simplificação por linha ou coluna completa é maior do que a simples eliminação dos termos indiferentes pela substituição pelo secundário. Levando-se em conta que o indiferente (irrelevante para a primeira sub-função) pode ser substituído pelo dominante ou pelo secundário, deve-se fazer a escolha que proporcione uma possível formação de linha ou coluna completa para cada ocorrência de valores dominantes, ou seja, substituição pelo dominante. Caso isso não seja possível devido à ocorrência de um valor secundário na linha ou na coluna em questão, deve-se fazer a substituição pelo secundário. Cada ocorrência de valores irrelevantes é tratada de maneira particular.

	1ª	2ª
C_1	*c0**	*c1*
0	0	$1_{(0)}$
1	1	1
2	$X_{(2)}$	2

	1ª	2ª
C_2	*c1**	*c2*
0	$X_{(0)}$	0
1	1	$2_{(1)}$
2	2	2

	1ª	2ª
C_2	*c2**	*c0*
0	0	0
1	$X_{(1)}$	1
2	2	$0_{(2)}$

Tabela 3.39

1ª	*c0**	*c1**	*c2**
Dominante	0	1	2
Secundário	1	2	0
Indiferente	$X_{(2)}$	$X_{(0)}$	$X_{(1)}$

2ª	*c0*	*c1*	*c2*
Dominante	0	1	2
Secundário	1	2	0
Indiferente	$0_{(2)}$	$1_{(0)}$	$2_{(1)}$

Tabela 3.40

1^a	$c*$
Dominante	Dominante
Secundário	Secundário
Indiferente	Irrelevante

2^a	c
Dominante	Dominante
Secundário	Secundário
Indiferente	Dominante

Tabela 3.41

$$C_1 = c0^* \, \gamma \, c1 \quad (3.7a)$$

$$C_2 = c1^* \, \alpha \, c2 \quad (3.7b)$$

$$C_3 = c2^* \, \beta \, c0 \quad (3.7c)$$

A quantidade de sub-funções necessárias para expressar a função depende da quantidade de valores que ela fornece, como é mostrado a seguir:

Valores	*c0*	*c1*	*c2*	Sub-funções
0 ou (0,1)	Sim			1
1 ou (1,2)		Sim		1
2 ou (2,0)			Sim	1
0,1,2	Sim	Sim	Sim	2

Tabela 3.42 – Sub-funções necessárias.

Exemplo:

		A		
	C	**0**	**1**	**2**
	0	0	0	0
B	**1**	0	1	1
	2	0	1	2

Tabela 3.43

Abaixo, é mostrada a solução sem simplificação da primeira sub-função.

	c0	A: 0	1	2
B	0	0	0	0
	1	0	1	1
	2	0	1	0

	c1	A: 0	1	2
B	0	1	1	1
	1	1	1	1
	2	1	1	2

	c2	A: 0	1	2
B	0	0	0	0
	1	0	2	2
	2	0	2	2

Tabela 3.44

$$c0 = (A\alpha 1)\alpha(B\alpha 1)\alpha(\overline{A}\beta\overline{B}) \qquad c0 = A\alpha B\alpha(\overline{A}\beta\overline{B})\alpha 1$$

$$c1 = (\overline{A}\beta 2)\beta(A\beta 2)\beta(\overline{B}\beta 2)\beta(B\beta 2) \qquad c1 = \overline{A}\beta A\beta\overline{B}\beta B\beta 2$$

$$c2 = (\overline{A}\alpha\overline{B})\gamma(A\alpha\overline{B})\gamma(\overline{A}\alpha B)\gamma(A\alpha B) \qquad c2 = (\overline{A}\alpha\overline{B})\gamma(A\alpha\overline{B})\gamma(\overline{A}\alpha B)\gamma(A\alpha B)$$

$$C_1 = c0\gamma c1 \qquad C_1 = (A\alpha B\alpha(\overline{A}\beta\overline{B})\alpha 1)\gamma(\overline{A}\beta A\beta\overline{B}\beta B\beta 2)$$

$$C_2 = c1\alpha c2 \qquad C_2 = (\overline{A}\beta A\beta\overline{B}\beta B\beta 2)\alpha((\overline{A}\alpha\overline{B})\gamma(A\alpha\overline{B})\gamma(\overline{A}\alpha B)\gamma(A\alpha B))$$

$$C_3 = c2\beta c0 \qquad C_3 = ((\overline{A}\alpha\overline{B})\gamma(A\alpha\overline{B})\gamma(\overline{A}\alpha B)\gamma(A\alpha B))\beta(A\alpha B\alpha(\overline{A}\beta\overline{B})\alpha 1)$$

Abaixo, é mostrada a solução com simplificação da primeira sub-função.

	$c0^*$	A: 0	1	2
B	0	0	0	0
	1	0	1	1
	2	0	1	1

	$c1^*$	A: 0	1	2
B	0	2	1	2
	1	1	1	1
	2	2	1	2

	$c2^*$	A: 0	1	2
B	0	0	0	0
	1	0	0	0
	2	0	0	2

Tabela 3.45

$$c0^* = (A\alpha 1)\alpha(B\alpha 1) \qquad c0^* = A\alpha B\alpha 1$$

$$c1^* = (A\beta 2)\beta(B\beta 2) \qquad c1^* = A\beta B\beta 2$$

$$c2^* = (A\alpha B)\gamma 0 \qquad c2^* = (A\alpha B)\gamma 0$$

$$C_1{}^* = c0^*\gamma c1 \qquad C_1 = (A\alpha B\alpha 1)\gamma(\overline{A}\beta A\beta\overline{B}\beta B\beta 2)$$

$$C_2{}^* = c1^*\alpha c2 \qquad C_2 = (A\beta B\beta 2)\alpha((\overline{A}\alpha\overline{B})\gamma(A\alpha\overline{B})\gamma(\overline{A}\alpha B)\gamma(A\alpha B))$$

$$C_3{}^* = c2^*\beta c0 \qquad C_3 = ((A\alpha B)\gamma 0)\beta(A\alpha B\alpha(\overline{A}\beta\overline{B})\alpha 1)$$

São, ao todo, $3^9 = 19683$ funções de três valores e duas entradas.

São, ao todo, $2^9 = 512$ funções binárias que podem ser utilizadas como sub-funções (sem considerar as três variações).

A escolha de qual modo utilizar (C_1, C_2, C_3) deve ser feita após a realização das três análises, pois é difícil prever qual deles sofrerá a maior redução devido às simplificações por linha e coluna completa.

3.5 Síntese de funções de quatro valores e uma entrada

Toda função de quatro valores pode ser decomposta em quatro sub-funções binárias. No caso de função de uma entrada *B=B(A)*, as sub-funções são *b0,b1,b2,b3*. Cada uma dessas funções fornece apenas dois valores, sendo um deles o valor a ser analisado (dominante), valor para o qual cada ocorrência fornece um termo.

Sub-função binária	Dominante Verdadeiro	Secundário Falso	Operação entre termos
b0	0	1	α
b1	1	2	β
b2	2	3	γ
b3	3	0	δ

Tabela 3.46 – Operações entre termos.

A função *B=B(A)* pode ser obtida de quatro formas:

$B_1=B_1(A)$ (3.8*a*)	$B_1=B_1(b0,b1,b2)$ (3.8*e*)	$b0=b0(A)$ (3.8*i*)
$B_2=B_2(A)$ (3.8*b*)	$B_2=B_2(b1,b2,b3)$ (3.8*f*)	$b1=b1(A)$ (3.8*j*)
$B_3=B_3(A)$ (3.8*c*)	$B_3=B_3(b2,b3,b0)$ (3.8*g*)	$b2=b2(A)$ (3.8*k*)
$B_4=B_4(A)$ (3.8*d*)	$B_4=B_4(b3,b0,b1)$ (3.8*h*)	$b3=b3(A)$ (3.8*l*)

Para converter a função *B=B(A)* nas sub-funções *b0,b1,b2,b3*, faz-se a substituição do valor terciário e do valor indiferente pelo valor dominante.

Sub-função	Substituição
b0	2,3 → 0
b1	3,0 → 1
b2	0,1 → 2
b3	1,2 → 3

Tabela 3.47

	b0	*b1*	*b2*	*b3*
0	0	$1_{(0)}$	$2_{(0)}$	0
1	1	1	$2_{(1)}$	$3_{(1)}$
2	$0_{(2)}$	2	2	$3_{(2)}$
3	$0_{(3)}$	$1_{(3)}$	3	3

Tabela 3.48

	b0	*b1*	*b2*	*b3*
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	$0_{(2)}$	$1_{(3)}$	$2_{(0)}$	$3_{(1)}$
Indiferente	$0_{(3)}$	$1_{(0)}$	$2_{(1)}$	$3_{(2)}$

Tabela 3.49

Lembrando dos operadores para a realização da conversão para binário ou eliminação do valor terciário e do valor indiferente: ($A\alpha 1$, $A\beta 2$, $A\gamma 3$, $A\delta 0$), as funções equivalentes aos termos podem ser obtidas diretamente das tabelas abaixo:

b0- Zeros

0	1	2	3
$A\alpha 1$	$\underline{A}\alpha 1$	$\overline{\overline{A}}\alpha 1$	$\overline{A}\alpha 1$

b1- Uns

0	1	2	3
$\overline{A}\beta 2$	$A\beta 2$	$\underline{A}\beta 2$	$\overline{\overline{A}}\beta 2$

b2- Dois

0	1	2	3
$\overline{\overline{A}}\gamma 3$	$\overline{A}\gamma 3$	$A\gamma 3$	$\underline{A}\gamma 3$

b3- Três

0	1	2	3
$\underline{A}\delta 0$	$\overline{\overline{A}}\delta 0$	$\overline{A}\delta 0$	$A\delta 0$

Tabela 3.50 – Termos.

Quando houver três termos ou mais, não é necessário fazer a conversão para binário como mostra a tabela abaixo. Se houver dois termos não vizinhos, intercalados, também não é necessário fazer a conversão.

Termos	dominante	secundário	terciário	indiferente	c.p.b.
1	1	1	1	1	Sim
2*	2	1	1		Sim
2**	2	2			Não
3	3	1			Não
4	4				Não

*quando os dois termos forem vizinhos
**quando os dois termos forem intercalados

Tabela 3.51 – Necessidade de fazer a conversão para binário.

A função completa utiliza três sub-funções. Ela é dividida em duas partes, sendo a primeira uma função ternária e a segunda uma função binária. A função ternária é formada pela primeira e a segunda sub-funções, a função binária é formada pela terceira sub-função.

A função ternária utiliza os conectivos determinados pelas regras das funções ternárias vistas na seção anterior.

Forma	Sub-funções binárias	Analisam, respectivamente	Operações entre termos das sub-funções	Operação entre sub-funções	Dominante do operador
B_1	*(b0,b1)*	(0,1)	(α,β)	γ	2
B_2	*(b1,b2)*	(1,2)	(β,γ)	δ	3
B_3	*(b2,b3)*	(2,3)	(γ,δ)	α	0
B_4	*(b3,b0)*	(3,0)	(δ,α)	β	1

Tabela 3.52 – Operações entre sub-funções binárias (função ternária).

A função completa utiliza, entre as partes ternária e binária, o conectivo para o qual os valores analisados nas três sub-funções sejam o secundário, o terciário e o indiferente, respectivamente, ou seja:

Função completa

Forma	Sub-funções binárias	Analisam, respectivamente	Operações entre termos das sub-funções	Operação entre sub-funções	Dominante do operador
B_1	*((b0,b1),b2)*	(0,1,2)	(α,β,γ)	δ	3
B_2	*((b1,b2),b3)*	(1,2,3)	(β,γ,δ)	α	0
B_3	*((b2,b3),b0)*	(2,3,0)	(γ,δ,α)	β	1
B_4	*((b3,b0),b1)*	(3,0,1)	(δ,α,β)	γ	2

Tabela 3.53 – Operações entre a sub-função ternária e a binária.

Sendo assim, as três formas da função são definidas como:

$$B_1 = (\,b0\ \gamma\, b1)\ \delta\, b2 \quad (3.9a)$$

$$B_2 = (\,b1\ \delta\, b2)\ \alpha\, b3 \quad (3.9b)$$

$$B_3 = (\,b2\ \alpha\, b3)\ \beta\, b0 \quad (3.9c)$$

$$B_4 = (\,b3\ \beta\, b0)\ \gamma\, b1 \quad (3.9d)$$

	1^a	2^a	3^a
B_1	*b0*	*b1*	*b2*
0	0	$1_{(0)}$	$2_{(0)}$
1	1	1	$2_{(1)}$
2	$0_{(2)}$	2	2
3	$0_{(3)}$	$1_{(3)}$	3

	1^a	2^a	3^a
B_2	*b1*	*b2*	*b3*
0	$1_{(0)}$	$2_{(0)}$	0
1	1	$2_{(1)}$	$3_{(1)}$
2	2	2	$3_{(2)}$
3	$1_{(3)}$	3	3

	1^a	2^a	3^a
B_3	*b2*	*b3*	*b0*
0	$2_{(0)}$	0	0
1	$2_{(1)}$	$3_{(1)}$	1
2	2	$3_{(2)}$	$0_{(2)}$
3	3	3	$0_{(3)}$

	1^a	2^a	3^a
B_4	*b3*	*b0*	*b1*
0	0	0	$1_{(0)}$
1	$3_{(1)}$	1	1
2	$3_{(2)}$	$0_{(2)}$	2
3	3	$0_{(3)}$	$1_{(3)}$

Tabela 3.54

Durante a análise, pode-se fazer a seguinte separação:

$B_1 = (...)\delta\, b2$ $\qquad$ $B_1 = (b0\ \gamma\, b1)...$

$B_2 = (...)\alpha\, b3$ $\qquad$ $B_2 = (b1\ \delta\, b2)...$

$B_3 = (...)\beta\, b0$ $\qquad$ $B_3 = (b2\ \alpha\, b3)...$

$B_4 = (...)\gamma\, b1$ $\qquad$ $B_4 = (b3\ \delta\, b0)...$

Na primeira coluna, percebe-se que o valor indiferente na primeira sub-função e o terciário na segunda sub-função são operados com o valor dominante na terceira sub-função. Por esse motivo, o valor indiferente na primeira sub-função e o terciário na segunda sub-função tornam-se irrelevantes. Como cada ocorrência do dominante acarreta em um termo na função, deve-se fazer a substituição pelo secundário.

	1^a	2^a	3^a
$B_1(\delta)$	*b0*	*b1*	*b2*
3	$0_{(3)}$	$1_{(3)}$	3

$X\delta 3 = 3$

	1^a	2^a	3^a
$B_2(\alpha)$	*b1*	*b2*	*b3*
0	$1_{(0)}$	$2_{(0)}$	0

$X\alpha 0 = 0$

	1^a	2^a	3^a
$B_3(\beta)$	*b2*	*b3*	*b0*
2	$3_{(1)}$	1	$2_{(1)}$

$X\beta 1 = 1$

	1^a	2^a	3^a
$B_4(\gamma)$	*b3*	*b0*	*b1*
2	$3_{(2)}$	$0_{(2)}$	2

$X\gamma 2 = 2$

Tabela 3.55

Na segunda coluna, percebe-se que o valor terciário na primeira sub-função é operado com o valor dominante na segunda sub-função. Por esse motivo, o valor terciário na primeira sub-função torna-se irrelevante. Deve-se fazer a substituição pelo secundário.

	1ª	2ª
$B_1(\gamma)$	$b0$	$b1$
2	$0_{(2)}$	2
$X\gamma 2 = 2$		

	1ª	2ª
$B_2(\delta)$	$b1$	$b2$
3	$1_{(3)}$	3
$X\delta 3 = 3$		

	1ª	2ª
$B_3(\alpha)$	$b2$	$b3$
0	$2_{(0)}$	0
$X\alpha 0 = 0$		

	1ª	2ª
$B_4(\beta)$	$b3$	$b0$
1	$3_{(1)}$	1
$X\beta 1 = 1$		

Tabela 3.56

Por convenção, quando a sub-função é obtida por meio da substituição do terciário e do indiferente por um irrelevante, ela é chamada de b^{**}. Quando a função é obtida da substituição do terciário pelo secundário e da substituição do indiferente por um irrelevante, ela é chamada de b^{*}.

	1ª	2ª	3ª
B_1	$b0^{**}$	$b1^{*}$	$b2$
0	0	$1_{(0)}$	$2_{(0)}$
1	1	1	$2_{(1)}$
2	$1_{(2)}$	2	2
3	$1_{(3)}$	$2_{(3)}$	3

	1ª	2ª	3ª
B_2	$b1^{**}$	$b2^{*}$	$b3$
0	$2_{(0)}$	$3_{(0)}$	0
1	1	$2_{(1)}$	$3_{(1)}$
2	2	2	$3_{(2)}$
3	$2_{(3)}$	3	3

	1ª	2ª	3ª
B_3	$b2^{**}$	$b3^{*}$	$b0$
0	$3_{(0)}$	0	0
1	$3_{(1)}$	$0_{(1)}$	1
2	2	$3_{(2)}$	$0_{(2)}$
3	3	3	$0_{(3)}$

	1ª	2ª	3ª
B_4	$b3^{**}$	$b0^{*}$	$b1$
0	0	0	$1_{(0)}$
1	$0_{(1)}$	1	1
2	$0_{(2)}$	$1_{(2)}$	2
3	3	$0_{(3)}$	$1_{(3)}$

Tabela 3.57

1ª	$b0^{**}$	$b1^{**}$	$b2^{**}$	$b3^{**}$
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	$1_{(2)}$	$2_{(3)}$	$3_{(0)}$	$0_{(1)}$
Indiferente	$1_{(3)}$	$2_{(0)}$	$3_{(1)}$	$0_{(2)}$

2ª	$b0^{*}$	$b1^{*}$	$b2^{*}$	$b3^{*}$
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	$1_{(2)}$	$2_{(3)}$	$3_{(0)}$	$0_{(1)}$
Indiferente	$0_{(3)}$	$1_{(0)}$	$2_{(1)}$	$3_{(2)}$

3ª	$b0$	$b1$	$b2$	$b3$
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	$0_{(2)}$	$1_{(3)}$	$2_{(0)}$	$3_{(1)}$
Indiferente	$0_{(3)}$	$1_{(0)}$	$2_{(1)}$	$3_{(2)}$

Tabela 3.58

De uma maneira geral, tem-se:

	1ª	2ª	3ª
B_1	b^{**}	b^{*}	b
0	1º	1º (4o)	1º (3o)
1	2º	1º	1º (4o)
2	2º (3o)	2º	1º
3	2º (4o)	2º (3o)	2º

	1ª	2ª	3ª
	b^{**}	b^{*}	b
Dominante	1º	1º	1º
Secundário	2º	2º	2º
Terciário	2º (3o)	2º (3o)	1º (3o)
Indiferente	2º (4o)	1º (4o)	1º (4o)

Tabela 3.59

1ª	b^{**}
Dominante	Dominante
Secundário	Secundário
Terciário	Secundário
Indiferente	Secundário

2ª	b^{*}
Dominante	Dominante
Secundário	Secundário
Terciário	Secundário
Indiferente	Dominante

3ª	b
Dominante	Dominante
Secundário	Secundário
Terciário	Dominante
Indiferente	Dominante

Tabela 3.60

$$B_1 = (\, b0^{**}\ \gamma\, b1^{*}\,)\ \delta\, b2 \quad (3.10a)$$

$$B_2 = (\, b1^{**}\ \delta\, b2^{*}\,)\ \alpha\, b3 \quad (3.10b)$$

$$B_3 = (\, b2^{**}\ \alpha\, b3^{*}\,)\ \beta\, b0 \quad (3.10b)$$

$$B_4 = (\, b3^{**}\ \beta\, b0^{*}\,)\ \gamma\, b1 \quad (3.10c)$$

A quantidade de sub-funções necessárias para expressar a função depende da quantidade de valores que ela fornece, como é mostrado abaixo:

Valores	*b0*	*b1*	*b2*	*b3*	Sub-funções
0 ou (0,1)	Sim				1
1 ou (1,2)		Sim			1
2 ou (2,3)			Sim		1
3 ou (3,0)				Sim	1
0,1,2,3	Sim	Sim	Sim	Sim	3

Tabela 3.61 – Sub-funções necessárias.

Será mostrada, agora, a análise de algumas das $4^4 = 256$ funções de quatro valores e uma entrada por ordem de simplicidade.

Uma vez que são usadas sub-funções binárias, podemos tabelar tais funções e utilizá-las na criação da função completa. São $2^4 = 16$ sub-funções binárias. Se se considerar as quatro variações (0,1), (1,2), (2,3), (3,0), passa-se para 64 sub-funções.

B	*b0*	*B*	*b0*
0000	0	1000	$\overline{A}\alpha\overline{\overline{A}}\alpha\underline{A}$
0001	$A\alpha\overline{A}\alpha\overline{\overline{A}}$	1001	$\overline{\overline{A}}\alpha\underline{A}\alpha 1$
0010	$\underline{A}\alpha A\alpha\overline{A}$	1010	$\overline{A}\alpha\underline{A}$
0011	$\underline{A}\alpha A\alpha 1$	1011	$\underline{A}\alpha 1$
0100	$A\alpha\overline{A}\alpha\overline{\overline{A}}$	1100	$\overline{A}\alpha\overline{\overline{A}}\alpha 1$
0101	$A\alpha\overline{\overline{A}}$	1101	$\overline{\overline{A}}\alpha 1$
0110	$A\alpha\overline{A}\alpha 1$	1110	$\overline{A}\alpha 1$
0111	$A\alpha 1$	1111	1

Tabela 3.62 – Exemplo de *b0*.

3.5.1- Funções equivalentes a binárias

As funções mostradas a seguir fornecem apenas o valor dominante e o secundário do conectivo utilizado. Estas funções correspondem a funções binárias e, por isso, requerem o uso de apenas uma sub-função. Estas são as 56 funções básicas e são utilizadas na síntese de funções que envolvam mais do que dois valores ou dois valores não consecutivos.

B	*b0*	*B*	*b1*	*B*	*b2*	*B*	*b3*
0111	$A\alpha 1$	2122	$A\beta 2$	3323	$A\gamma 3$	0003	$A\delta 0$
1110	$\overline{A}\alpha 1$	1222	$\overline{A}\beta 2$	3233	$\overline{A}\gamma 3$	0030	$\overline{A}\delta 0$
1101	$\overline{\overline{A}}\alpha 1$	2221	$\overline{\overline{A}}\beta 2$	3233	$\overline{\overline{A}}\gamma 3$	0300	$\overline{\overline{A}}\delta 0$
1011	$\underline{A}\alpha 1$	2212	$\underline{A}\beta 2$	2333	$\underline{A}\gamma 3$	3000	$\underline{A}\delta 0$
0110	$A\alpha\overline{A}\alpha 1$	1122	$A\beta\overline{A}\beta 2$	3223	$A\gamma\overline{A}\gamma 3$	0033	$A\delta\overline{A}\delta 0$
1100	$\overline{A}\alpha\overline{\overline{A}}\alpha 1$	1221	$\overline{A}\beta\overline{\overline{A}}\beta 2$	2233	$\overline{A}\gamma\overline{\overline{A}}\gamma 3$	0330	$\overline{A}\delta\overline{\overline{A}}\delta 0$
1001	$\overline{\overline{A}}\alpha\underline{A}\alpha 1$	2211	$\overline{\overline{A}}\beta\underline{A}\beta 2$	2332	$\overline{\overline{A}}\gamma\underline{A}\gamma 3$	3300	$\overline{\overline{A}}\delta\underline{A}\delta 0$
0011	$\underline{A}\alpha A\alpha 1$	2112	$\underline{A}\beta A\beta 2$	3322	$\underline{A}\gamma A\gamma 3$	3003	$\underline{A}\delta A\delta 0$
0101	$A\alpha\overline{\overline{A}}$	2121	$A\beta\overline{\overline{A}}$	2323	$A\gamma\overline{\overline{A}}$	0303	$A\delta\overline{\overline{A}}$
1010	$\overline{A}\alpha\underline{A}$	1212	$\overline{A}\beta\underline{A}$	3232	$\overline{A}\gamma\underline{A}$	3030	$\overline{A}\delta\underline{A}$
0100	$A\alpha\overline{A}\alpha\overline{\overline{A}}$	1121	$A\beta\overline{A}\beta\overline{\overline{A}}$	2223	$A\gamma\overline{A}\gamma\overline{\overline{A}}$	0333	$A\delta\overline{A}\delta\overline{\overline{A}}$
1000	$\overline{A}\alpha\overline{\overline{A}}\alpha\underline{A}$	1211	$\overline{A}\beta\overline{\overline{A}}\beta\underline{A}$	2232	$\overline{A}\gamma\overline{\overline{A}}\gamma\underline{A}$	3330	$\overline{A}\delta\overline{\overline{A}}\delta\underline{A}$
0010	$\underline{A}\alpha A\alpha\overline{A}$	2111	$\overline{\overline{A}}\beta\underline{A}\beta A$	2322	$\overline{\overline{A}}\gamma\underline{A}\gamma A$	3303	$\overline{\overline{A}}\delta\underline{A}\delta A$
0001	$A\alpha\overline{A}\alpha\overline{\overline{A}}$	1112	$\underline{A}\beta A\beta\overline{A}$	3222	$\underline{A}\gamma A\gamma\overline{A}$	3033	$\underline{A}\delta A\delta\overline{A}$

Tabela 3.63 – As 56 funções básicas.

3.5.2- Funções com os quatro valores

Exemplo 1:

$b0$: (2,3) → 0

$b1$: (3,0) → 1

$b2$: (0,1) → 2

$b3$: (1,2) → 3

A	**0**	**1**	**2**	**3**
B	**0**	**1**	**2**	**3**
b0	0	1	0	0
b1	1	1	2	1
b2	2	2	2	3
b2	0	3	3	3

$$b0 = A\alpha\overline{A}\alpha\overline{\overline{A}}$$
$$b1 = A\beta\overline{A}\beta\overline{\overline{A}}$$
$$b2 = A\gamma\overline{A}\gamma\overline{\overline{A}}$$
$$b3 = A\delta\overline{A}\delta\overline{\overline{A}}$$

Tabela 3.64

$$B_1 = (b0\gamma b1)\delta b2 \qquad B_1 = \left(\left(A\alpha\overline{A}\alpha\overline{\overline{A}}\right)\gamma\left(A\beta\overline{A}\beta\overline{\overline{A}}\right)\right)\delta\left(A\gamma\overline{A}\gamma\overline{\overline{A}}\right)$$
$$B_2 = (b1\delta b2)\alpha b3 \qquad B_2 = \left(\left(A\beta\overline{A}\beta\overline{\overline{A}}\right)\delta\left(A\gamma\overline{A}\gamma\overline{\overline{A}}\right)\right)\alpha\left(A\delta\overline{A}\delta\overline{\overline{A}}\right)$$
$$B_3 = (b2\alpha b3)\beta b0 \qquad B_3 = \left(\left(A\gamma\overline{A}\gamma\overline{\overline{A}}\right)\alpha\left(A\delta\overline{A}\delta\overline{\overline{A}}\right)\right)\beta\left(A\alpha\overline{A}\alpha\overline{\overline{A}}\right)$$
$$B_4 = (b3\beta b0)\gamma b1 \qquad B_4 = \left(\left(A\delta\overline{A}\delta\overline{\overline{A}}\right)\beta\left(A\alpha\overline{A}\alpha\overline{\overline{A}}\right)\right)\gamma\left(A\beta\overline{A}\beta\overline{\overline{A}}\right)$$

Utilizando irrelevantes:

$b0^{**}$:	$(2,3) \rightarrow 1$	0	1	1	1	$b0^{**} = A\alpha 1$
$b1^{**}$:	$(3,0) \rightarrow 2$	2	1	2	2	$b1^{**} = A\beta 2$
$b2^{**}$:	$(0,1) \rightarrow 3$	3	3	2	3	$b2^{**} = A\gamma 3$
$b3^{**}$:	$(1,2) \rightarrow 0$	0	0	0	3	$b3^{**} = A\delta 0$
$b0^{*}$:	$(2\rightarrow 1)$, $(3\rightarrow 0)$	0	1	1	0	$b0^{*} = A\alpha \overline{A}\alpha 1$
$b1^{*}$:	$(3\rightarrow 2)$, $(0\rightarrow 1)$	1	1	2	2	$b1^{*} = A\beta \overline{A}\beta 2$
$b2^{*}$:	$(0\rightarrow 3)$, $(1\rightarrow 2)$	3	2	2	3	$b2^{*} = A\gamma \overline{A}\gamma 3$
$b3^{*}$:	$(1\rightarrow 0)$, $(2\rightarrow 3)$	0	0	3	3	$b3^{*} = A\delta \overline{A}\delta 0$

$$B_1^{*} = \left(b0^{**}\gamma b1^{*}\right)\delta b2 \qquad B_1 = \left(\left(A\alpha 1\right)\gamma\left(A\beta\overline{A}\beta 2\right)\right)\delta\left(A\gamma\overline{A}\gamma\overline{\overline{A}}\right)$$

$$B_2^{*} = \left(b1^{**}\delta b2^{*}\right)\alpha b3 \qquad B_2 = \left(\left(A\beta 2\right)\delta\left(A\gamma\overline{A}\gamma 3\right)\right)\alpha\left(A\delta\overline{A}\delta\overline{\overline{A}}\right)$$

$$B_3^{*} = \left(b2^{**}\alpha b3^{*}\right)\beta b0 \qquad B_3 = \left(\left(A\gamma 3\right)\alpha\left(A\delta\overline{A}\delta 0\right)\right)\beta\left(A\alpha\overline{A}\alpha\overline{\overline{A}}\right)$$

$$B_4^{*} = \left(b3^{**}\beta b0^{*}\right)\gamma b1 \qquad B_4 = \left(\left(A\delta 0\right)\beta\left(A\alpha\overline{A}\alpha 1\right)\right)\gamma\left(A\beta\overline{A}\beta\overline{\overline{A}}\right)$$

Exemplo 2:

		A	**0**	**1**	**2**	**3**	
		B	**3**	**2**	**1**	**0**	
$b0$:	$(2,3) \rightarrow 0$	*b0*	0	0	1	0	$b0 = \underline{A}\alpha A\alpha\overline{A}$
$b1$:	$(3,0) \rightarrow 1$	*b1*	1	2	1	1	$b1 = \overline{A}\beta\overline{\overline{A}}\beta A$
$b2$:	$(0,1) \rightarrow 2$	*b2*	3	2	2	2	$b2 = \underline{A}\gamma A\gamma\overline{A}$
$b3$:	$(1,2) \rightarrow 3$	*b2*	3	3	3	0	$b3 = \overline{A}\delta\overline{\overline{A}}\delta A$

Tabela 3.65

$$B_1 = \left(\left(\underline{A}\alpha A\alpha\overline{A}\right)\gamma\left(\overline{A}\beta\overline{\overline{A}}\beta\underline{A}\right)\right)\delta\left(\underline{A}\gamma A\gamma\overline{A}\right)$$

$$B_2 = \left(\left(\overline{A}\beta\overline{\overline{A}}\beta\underline{A}\right)\delta\left(\underline{A}\gamma A\gamma\overline{A}\right)\right)\alpha\left(\overline{A}\delta\overline{\overline{A}}\delta\underline{A}\right)$$

$$B_3 = \left(\left(\underline{A}\gamma A\gamma\overline{A}\right)\alpha\left(\overline{A}\delta\overline{\overline{A}}\delta\underline{A}\right)\right)\beta\left(\underline{A}\alpha A\alpha\overline{A}\right)$$

$$B_4 = \left(\left(\overline{A}\delta\overline{\overline{A}}\delta\underline{A}\right)\beta\left(\underline{A}\alpha A\alpha\overline{A}\right)\right)\gamma\left(\overline{A}\beta\overline{\overline{A}}\beta\underline{A}\right)$$

3.5.3- Funções equivalentes a ternárias

Se a função fornecer apenas três valores, pode-se utilizar apenas duas sub-funções, como mostra a tabela abaixo. Trata-se de um caso semelhante ao da lógica de três valores. Neste caso, a sub-função b^* continua válida, pois o valor terciário em quatro valores corresponde ao indiferente em três valores, sendo eliminado o último dos quatro valores.

	1ª	2ª	3ª
B_1	b^{**}	b^*	b
0	1º	1º (4o)	-
1	2º	1º	-
2	2º (3o)	2º	-
3	-	-	-

Tabela 3.66

Valores	Sub-funções	Forma
(0,1,2)	*(b0,b1)*	B_1
(1,2,3)	*(b1,b2)*	B_2
(2,3,0)	*(b2,b3)*	B_3
(3,0,1)	*(b3,b0)*	B_4

Tabela 3.67

Exemplo:

A	**0**	**1**	**2**	**3**
B	**0**	**1**	**2**	**2**
b0	0	1	0	0
b1	1	1	2	2

$$b0 = A\alpha\overline{A}\alpha\overline{\overline{A}}$$
$$b1 = A\beta\overline{A}\beta 2$$

Tabela 3.68a

A	**0**	**1**	**2**	**3**
B	**0**	**1**	**2**	**2**
$b0^*$	0	1	1	1

$$b0^* = A\alpha 1$$

Tabela 3.68b

$$B_1 = b0\gamma b1 \qquad B_1 = (A\alpha\overline{A}\alpha\overline{\overline{A}})\gamma(A\beta\overline{A}\beta 2)$$

$$B_1{}^* = b0^*\gamma b1 \qquad B_1{}^* = (A\alpha 1)\gamma(A\beta\overline{A}\beta 2)$$

3.5.4- Funções com dois valores não vizinhos

Se a função fornecer apenas dois valores não vizinhos, não se pode fazer a análise binária, pois ela não é uma função binária, apesar de conter apenas dois valores. Pode-se, no entanto, utilizar apenas duas sub-funções, como mostra a tabela abaixo. Tem-se um caso semelhante ao da lógica de três valores. Trata-se, então, de uma função ternária. Entretanto, ao contrário da função ternária, que só pode ser tratada de uma forma na lógica de quatro valores, esta função pode ser tratada de duas formas:

Valores	Sub-funções	Forma	3 valores	Valores	Sub-funções	Forma	3 valores
0,2	*b0,b1* *b2,b3*	B_1 B_3	0,1,2 2,3,0	2,0	*b2,b3* *b0b1*	B_3 B_1	2,3,0 0,1,2
1,3	*b1,b2* *b3b0*	B_2 B_4	1,2,3 3,0,1	3,1	*b3,b0* *b1b2*	B_4 B_2	3,0,1 1,2,3

Tabela 3.69

Exemplo:

A	**0**	**1**	**2**	**3**	
B	**0**	**2**	**0**	**2**	
b0	0	0	0	0	$b0 = 0$
b1	1	2	1	2	$b1 = \overline{A}\beta\underline{A}$
b2	2	2	2	2	$b2 = 2$
b3	0	3	0	3	$b3 = A\delta\overline{\overline{A}}$

Tabela 3.70

$$B_1 = b0\gamma b1 \qquad B_1 = \left(\overline{A}\beta\underline{A}\right)\gamma 0$$

$$B_3 = b2\alpha b3 \qquad B_3 = \left(A\delta\overline{\overline{A}}\right)\alpha 2$$

Percebe-se que as sub-funções *b1* e *b3* sofreram a conversão para ternário, fornecendo as funções completas.

Nota-se que a fórmula completa continua sendo válida, embora redundante, pois temos uma operação com o indiferente:

$$B_1 = (b0\gamma b1)\delta b2 = (b0\gamma b1)\delta 2 = b0\gamma b1$$
$$B_3 = (b2\alpha b3)\beta b0 = (b2\alpha b3)\beta 0 = b2\alpha b3$$

O uso de irrelevantes, neste caso, não oferece vantagem, como é mostrado abaixo.

A	**0**	**1**	**2**	**3**
B	**0**	**2**	**0**	**2**
*b0**	0	1	0	1
b1	1	2	1	2
*b2**	3	2	3	2
b3	0	3	0	3

$$b0^* = A\alpha \overline{\overline{A}}$$
$$b1 = \overline{A}\beta \underline{A}$$
$$b2^* = \overline{A}\gamma \underline{A}$$
$$b3 = A\delta \overline{\overline{A}}$$

Tabela 3.71

$$B_1 = b0^* \gamma b1 \qquad B_1 = \left(A\alpha \overline{\overline{A}}\right)\gamma\left(\overline{A}\beta \underline{A}\right)$$
$$B_3 = b2^* \alpha b3 \qquad B_3 = \left(\overline{A}\gamma \underline{A}\right)\alpha\left(A\delta \overline{\overline{A}}\right)$$

A quantidade de sub-funções para expressar a função depende da quantidade de valores que ela fornece, como é mostrado abaixo:

Valores	*b0*	*b1*	*b2*	*b3*	Sub-funções	Formas	Redução
0 ou (0,1)	Sim				1	B_1	Binário
1 ou (1,2)		Sim			1	B_2	Binário
2 ou (2,3)			Sim		1	B_3	Binário
3 ou (3,0)				Sim	1	B_4	Binário
(0,1,2)	Sim	Sim			2	B_1	Ternário
(1,2,3)		Sim	Sim		2	B_2	Ternário
(2,3,0)			Sim	Sim	2	B_3	Ternário
(3,0,1)	Sim			Sim	2	B_4	Ternário
(0,2)	Sim	Sim	Sim	Sim	2	B_1, B_3	Ternário
(1,3)	Sim	Sim	Sim	Sim	2	B_2, B_4	Ternário
0,1,2,3	Sim	Sim	Sim	Sim	3	B_1, B_2, B_3, B_4	Quaternário

Tabela 3.72 – Sub-funções necessárias para todas as possibilidades.

3.6 Síntese de funções de quatro valores e duas entradas

Todas as regras abordadas para síntese de funções de quatro valores e uma entrada são aplicáveis à síntese de funções de quatro valores e duas entradas. A função é *C=C(A,B)*, as sub-funções são *c0,c1,c2,c3*. Se a função analisada fornecer apenas dois valores consecutivos, ela será fornecida diretamente por uma das quatro sub-funções *c0,c1,c2,c3*, não sendo necessária a análise das outras três sub-funções.

A função *C=C(A,B)* pode ser obtida de quatro formas:

$C_1=C_1(A,B)$ (3.11*a*) $C_1=C_1(c0,c1,c2)$ (3.11*f*) $c0=c0(A,B)$ (3.11*j*)

$C_2=C_2(A,B)$ (3.11*b*) $C_2=C_2(c1,c2,c3)$ (3.11*g*) $c1=c1(A,B)$ (3.11*k*)

$C_3=C_3(A,B)$ (3.11*c*) $C_3=C_3(c2,c3,c0)$ (3.11*h*) $c2=c2(A,B)$ (3.11*l*)

$C_4=C_4(A,B)$ (3.11*e*) $C_4=C_4(c3,c0,c1)$ (3.11*i*) $c3=c3(A,B)$ (3.11*m*)

$$C_1 = (c0\ \gamma\ c1)\ \delta\ c2 \quad (3.11n)$$

$$C_2 = (c1\ \delta\ c2)\ \alpha\ c3 \quad (3.11o)$$

$$C_3 = (c2\ \alpha\ c3)\ \beta\ c0 \quad (3.11p)$$

$$C_4 = (c3\ \beta\ c0)\ \gamma\ c1 \quad (3.11q)$$

Para converter a função *C=C(A,B)* nas sub-funções *c0,c1,c2,c3*, faz-se a substituição do valor terciário e do valor indiferente pelo valor dominante.

Sub-função	Substituição
c0	$2,3 \rightarrow 0$
c1	$3,0 \rightarrow 1$
c2	$0,1 \rightarrow 2$
c3	$1,2 \rightarrow 3$

Tabela 3.73

	c0	*c1*	*c2*	*c3*
0	0	$1_{(0)}$	$2_{(0)}$	0
1	1	1	$2_{(1)}$	$3_{(1)}$
2	$0_{(2)}$	2	2	$3_{(2)}$
3	$0_{(3)}$	$1_{(3)}$	3	3

Tabela 3.74

	c0	*c1*	*c2*	*c3*
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	$0_{(2)}$	$1_{(3)}$	$2_{(0)}$	$3_{(1)}$
Indiferente	$0_{(3)}$	$1_{(0)}$	$2_{(1)}$	$3_{(2)}$

Tabela 3.75

As expressões equivalentes a linhas e colunas completas podem ser obtidas diretamente das tabelas mostradas abaixo, provenientes dos termos das funções de uma entrada, tal como foi visto para as funções de três valores.

c0 - Zeros

0	1	2	3
$A\alpha 1$	$\underline{A}\alpha 1$	$\overline{\overline{A}}\alpha 1$	$\overline{A}\alpha 1$

c1 - Uns

0	1	2	3
$\overline{A}\beta 2$	$A\beta 2$	$\underline{A}\beta 2$	$\overline{\overline{A}}\beta 2$

c2 - Dois

0	1	2	3
$\overline{\overline{A}}\gamma 3$	$\overline{A}\gamma 3$	$A\gamma 3$	$\underline{A}\gamma 3$

c3 - Três

0	1	2	3
$\underline{A}\delta 0$	$\overline{\overline{A}}\delta 0$	$\overline{A}\delta 0$	$A\delta 0$

Tabela 3.76 – Termos equivalentes a colunas completas.

Foram mostrados os termos para coluna completa. Para linha completa, basta substituir a variável *A* pela variável *B*.

Um termo isolado, para duas entradas, faz parte de uma matriz bidimensional e, por isso, deve ser expresso em termos de suas coordenadas (*x*,*y*), que, no caso, são chamadas de (*A*,*B*). O valor das coordenadas é expresso em termos de deslocamentos.

Sub-função binária	Dominante Verdadeiro	Secundário Falso	Operação entre coordenadas	Operação entre termos
c0	0	1	β	α
c1	1	2	γ	β
c2	2	3	δ	γ
c3	3	0	α	δ

Tabela 3.77 – Operações entre coordenadas e entre termos.

Os conectivos a serem utilizados entre as sub-funções são determinados pelos mesmos critérios empregados na síntese de funções de uma entrada.

Forma	Operação entre as duas primeiras sub-funções binárias	Operação entre a sub-função ternária e a terceira sub-função binária
C_1	γ	δ
C_2	δ	α
C_3	α	β
C_4	β	γ

Tabela 3.78

Sendo assim, é possível resumir o uso das operações usadas por cada sub-função e função. O símbolo “ ”, utilizado na tabela abaixo, representa qualquer variável de entrada deslocada de qualquer valor $\left(A, \overline{A}, \overline{\overline{A}}, \underline{A}, B, \overline{B}, \overline{\overline{B}}, \underline{B}\right)$.

	Operação entre coordenadas	Operação entre termos	Aspecto:
c0	$\beta(1)$	$\alpha(0)$	$(\quad \beta \quad)\alpha(\quad \beta \quad)$
c1	$\gamma(2)$	$\beta(1)$	$(\quad \gamma \quad)\beta(\quad \gamma \quad)$
c2	$\delta(3)$	$\gamma(2)$	$(\quad \delta \quad)\gamma(\quad \delta \quad)$
c3	$\alpha(0)$	$\delta(3)$	$(\quad \alpha \quad)\delta(\quad \alpha \quad)$

Tabela 3.79

Termos isolados podem ser obtidos a partir da tabela mostrada a seguir. Para não sobrecarregar a tabela, a conversão para binário foi omitida; no entanto, ela sempre deve ser levada em consideração ($A\alpha 1$, $A\beta 2$, $A\gamma 3$, $A\delta 0$).

c0 - Zeros

0	*A = 0*	*A = 1*	*A = 2*	*A = 3*
B = 0	$A\beta B$	$\underline{A}\beta B$	$\overline{\overline{A}}\beta B$	$\overline{A}\beta B$
B = 1	$A\beta \underline{B}$	$\underline{A}\beta \underline{B}$	$\overline{\overline{A}}\beta \underline{B}$	$\overline{A}\beta \underline{B}$
B = 2	$A\beta \overline{\overline{B}}$	$\underline{A}\beta \overline{\overline{B}}$	$\overline{\overline{A}}\beta \overline{\overline{B}}$	$\overline{A}\beta \overline{\overline{B}}$
B=3	$A\beta \overline{B}$	$\underline{A}\beta \overline{B}$	$\overline{\overline{A}}\beta \overline{B}$	$\overline{A}\beta \overline{B}$

c1 - Uns

1	*A = 0*	*A = 1*	*A = 2*	*A = 3*
B = 0	$\overline{A}\gamma \overline{B}$	$A\gamma \overline{B}$	$\underline{A}\gamma \overline{B}$	$\overline{\overline{A}}\gamma \overline{B}$
B = 1	$\overline{A}\beta B$	$A\lambda B$	$\underline{A}\lambda B$	$\overline{\overline{A}}\beta B$
B = 2	$\overline{A}\gamma \underline{B}$	$A\lambda \underline{B}$	$\underline{A}\lambda \underline{B}$	$\overline{\overline{A}}\lambda \underline{B}$
B=3	$\overline{A}\gamma \overline{\overline{B}}$	$A\lambda \overline{\overline{B}}$	$\underline{A}\lambda \overline{\overline{B}}$	$\overline{\overline{A}}\lambda \overline{\overline{B}}$

c2 - Dois

2	*A = 0*	*A = 1*	*A = 2*	*A = 3*
B = 0	$\overline{\overline{A}}\delta \overline{\overline{B}}$	$\overline{A}\delta \overline{\overline{B}}$	$A\delta \overline{\overline{B}}$	$\underline{A}\delta \overline{\overline{B}}$
B = 1	$\overline{\overline{A}}\delta \overline{B}$	$\overline{A}\delta \overline{B}$	$A\delta \overline{B}$	$\underline{A}\delta \overline{B}$
B = 2	$\overline{\overline{A}}\delta B$	$\overline{A}\delta B$	$A\delta B$	$\underline{A}\delta B$
B=3	$\overline{\overline{A}}\delta \underline{B}$	$\overline{A}\delta \underline{B}$	$A\delta \underline{B}$	$\underline{A}\delta \underline{B}$

c3 - Três

3	*A = 0*	*A = 1*	*A = 2*	*A = 3*
B = 0	$\underline{A}\alpha \underline{B}$	$\overline{\overline{A}}\alpha \underline{B}$	$\overline{A}\alpha \underline{B}$	$A\alpha \underline{B}$
B = 1	$\underline{A}\alpha \overline{\overline{B}}$	$\overline{\overline{A}}\alpha \overline{\overline{B}}$	$\overline{A}\alpha \overline{\overline{B}}$	$A\alpha \overline{\overline{B}}$
B = 2	$\underline{A}\alpha \overline{B}$	$\overline{\overline{A}}\alpha \overline{B}$	$\overline{A}\alpha \overline{B}$	$A\alpha \overline{B}$
B=3	$\underline{A}\alpha B$	$\overline{\overline{A}}\alpha B$	$\overline{A}\alpha B$	$A\alpha B$

Tabela 3.80 – Termos.

Abaixo, estão alguns exemplos didáticos para *c0*.

Exemplo 1: Conectando-se os termos da tabela, tem-se $C = (A\beta B)\alpha(\underline{A}\beta \underline{B})$.

Fornecido

	A			
	0	1	1	1
B	1	**0**	1	1
	1	1	1	1
	1	1	1	1

$A\beta B$

	A			
	0	1	2	3
B	1	1	1	1
	2	1	2	2
	3	1	2	3

$\underline{A}\beta \underline{B}$

	A			
	3	3	1	2
B	3	**0**	1	2
	1	1	1	1
	2	2	1	2

$(A\beta B)\alpha(\underline{A}\beta \underline{B})$

	A			
	0	1	1	2
B	1	**0**	1	1
	1	1	1	1
	2	1	1	2

Tabela 3.81

Ao verificar esta função, percebe-se que ela não corresponde à função dada, deve-se fazer a conversão para binário. A propriedade “*Sempre que a função contiver pelo menos dois termos em linhas e colunas diferentes, a conversão para binário é dispensável*” vale apenas para síntese de funções de três valores, não é válida para quatro valores.

Exemplo 2: Conectando-se os termos da tabela tem-se $C = (A\beta B)\alpha(\overline{\overline{A}}\beta\overline{\overline{B}})$.

Fornecido

	A			
	0	1	1	1
B	1	1	1	1
	1	1	**0**	1
	1	1	1	1

$A\beta B$

	A			
	0	1	2	3
B	1	1	1	1
	2	1	2	2
	3	1	2	3

$\overline{\overline{A}}\beta\overline{\overline{B}}$

	A			
	2	2	2	1
B	2	3	3	1
	2	3	**0**	1
	1	1	1	1

$(A\beta B)\alpha(\overline{\overline{A}}\beta\overline{\overline{B}})$

	A			
	0	1	2	1
B	1	1	1	1
	2	1	**0**	1
	1	1	1	1

Tabela 3.82

Ao verificar esta função, percebe-se que ela não corresponde à função dada, deve-se fazer a conversão para binário. A propriedade "*Se tivermos dois termos não visinhos, em diagonal, intercalados, não é necessário fazer a conversão para binário.*" Aplica-se apenas em síntese de funções de uma entrada, não é válida para duas entradas.

Exemplo 3: Conectando-se os termos da tabela tem-se $C = (A\beta B)\alpha(\underline{A}\beta\underline{B})\alpha(\overline{\overline{A}}\beta\overline{\overline{B}})$.

Fornecido

	A			
	0	1	1	1
B	1	**0**	1	1
	1	1	**0**	1
	1	1	1	1

$A\beta B$

	A			
	0	1	2	3
B	1	1	1	1
	2	1	2	2
	3	1	2	3

$\underline{A}\beta\underline{B}$

	A			
	3	3	1	2
B	3	**0**	1	2
	1	1	1	1
	2	2	1	2

$\overline{\overline{A}}\beta\overline{\overline{B}}$

	A			
	2	2	2	1
B	2	3	3	1
	2	3	**0**	1
	1	1	1	1

$(A\beta B)\alpha(\underline{A}\beta\underline{B})\alpha(\overline{\overline{A}}\beta\overline{\overline{B}})$

	A			
	0	1	1	1
B	1	**0**	1	1
	1	1	**0**	1
	1	1	1	1

Tabela 3.83

Ao verificar esta função, percebe-se que ela corresponde à função dada. Sempre que a função contiver pelo menos três termos em linhas e colunas diferentes, em diagonal, a conversão para binário é dispensável.

Pode-se concluir que, da mesma forma que, para funções de uma entrada, a função (0001) não precisa da conversão para binário analisado os zeros, para duas entradas, a função $\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ também não precisa.

Exemplo 4:

Fornecido

	A			
B	**0**	1	**0**	1
	1	1	1	1
	0	1	1	1
	1	1	1	1

$A\beta B$

	A			
B	**0**	1	2	3
	1	1	1	1
	2	1	2	2
	3	1	2	3

$A\beta \overline{\overline{B}}$

	A			
B	2	1	2	2
	3	1	2	3
	0	1	2	3
	1	1	1	1

$\overline{\overline{A}}\beta B$

	A			
B	2	3	**0**	1
	1	1	1	1
	2	2	2	1
	2	3	3	1

C

	A			
B	**0**	1	**0**	1
	1	1	1	1
	0	1	2	1
	1	1	1	1

Tabela 3.84

Exemplo 5:

Fornecido

	A			
B	**0**	1	**0**	1
	1	1	1	1
	0	1	**0**	1
	1	1	1	1

$A\beta B$

	A			
B	**0**	1	2	3
	1	1	1	1
	2	1	2	2
	3	1	2	3

$A\beta \overline{\overline{B}}$

	A			
B	2	1	2	2
	3	1	2	3
	0	1	2	3
	1	1	1	1

$\overline{\overline{A}}\beta B$

	A			
B	2	3	**0**	1
	1	1	1	1
	2	2	2	1
	2	3	3	1

$\overline{\overline{A}}\beta \overline{\overline{B}}$

	A			
B	2	2	2	1
	2	3	3	1
	2	3	**0**	1
	1	1	1	1

C

	A			
B	**0**	1	**0**	1
	1	1	1	1
	0	1	**0**	1
	1	1	1	1

Tabela 3.85

Exemplo 6:

Fornecido

	A			
B	**0**	**0**	1	1
	0	**0**	1	1
	1	1	1	1
	1	1	1	1

$A\beta B$

	A			
B	**0**	1	2	3
	1	1	1	1
	2	1	2	2
	3	1	2	3

$A\beta \underline{B}$

	A			
B	3	1	2	3
	0	1	2	3
	1	1	1	1
	2	1	2	2

$\underline{A}\beta B$

	A			
B	3	**0**	1	2
	1	1	1	1
	2	2	1	2
	3	3	1	2

$\underline{A}\beta \underline{B}$

	A			
B	3	3	1	2
	3	**0**	1	2
	1	1	1	1
	2	2	1	2

C

	A			
B	**0**	**0**	1	2
	0	**0**	1	1
	1	1	1	1
	2	1	1	2

Tabela 3.86

Pode-se concluir que, da mesma forma que, para funções de uma entrada, a função (0011) precisa da conversão para binário e a função (0101) não precisa, a função $\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ precisa da conversão para binário e a função $\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ não precisa.

Levando-se em conta que o valor indiferente (irrelevante para a primeira sub-função) pode ser substituído pelo dominante ou pelo secundário, deve-se fazer a escolha que proporcione uma possível formação de linha ou coluna completa para cada ocorrência de valores dominantes, como foi visto para três valores.

	1ª	2ª	3ª
C_1	$c0^{**}$	$c1^{*}$	$c2$
0	0	$1_{(0)}$	$2_{(0)}$
1	1	1	$2_{(1)}$
2	$X_{(2)}$	2	2
3	$X_{(3)}$	$X_{(3)}$	3

	1ª	2ª	3ª
C_2	$c1^{**}$	$c2^{*}$	$c3$
0	$X_{(0)}$	$X_{(0)}$	0
1	1	$2_{(1)}$	$3_{(1)}$
2	2	2	$3_{(2)}$
3	$X_{(3)}$	3	3

	1ª	2ª	3ª
C_3	$c2^{**}$	$c3^{*}$	$c0$
0	$X_{(0)}$	0	0
1	$X_{(1)}$	$X_{(1)}$	1
2	2	$3_{(2)}$	$0_{(2)}$
3	3	3	$0_{(3)}$

	1ª	2ª	3ª
C_4	$c3^{**}$	$c0^{*}$	$c1$
0	0	0	$1_{(0)}$
1	$X_{(1)}$	1	1
2	$X_{(2)}$	$X_{(2)}$	2
3	3	$0_{(3)}$	$1_{(3)}$

Tabela 87

1ª	$c0^{**}$	$c1^{**}$	$c2^{**}$	$c3^{**}$
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	X	X	X	X
Indiferente	X	X	X	X

2ª	$c0^{*}$	$c1^{*}$	$c2^{*}$	$c3^{*}$
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	$X_{(2)}$	$X_{(3)}$	$X_{(0)}$	$X_{(1)}$
Indiferente	$0_{(3)}$	$1_{(0)}$	$2_{(1)}$	$3_{(2)}$

3ª	$c0$	$c1$	$c2$	$c3$
Dominante	0	1	2	3
Secundário	1	2	3	0
Terciário	$0_{(2)}$	$1_{(3)}$	$2_{(0)}$	$3_{(1)}$
Indiferente	$0_{(3)}$	$1_{(0)}$	$2_{(1)}$	$3_{(2)}$

Tabela 88

1ª	c^{**}
Dominante	Dominante
Secundário	Secundário
Terciário	Irrelevante
Indiferente	Irrelevante

2ª	c^{*}
Dominante	Dominante
Secundário	Secundário
Terciário	Irrelevante
Indiferente	Dominante

3ª	c
Dominante	Dominante
Secundário	Secundário
Terciário	Dominante
Indiferente	Dominante

Tabela 89

$$C_1 = (c0^{**}\ \gamma\, c1^{*})\ \delta\, c2 \quad (3.12a)$$

$$C_2 = (c1^{**}\ \delta\, c2^{*})\ \alpha\, c3 \quad (3.12b)$$

$$C_3 = (c2^{**}\ \alpha\, c3^{*})\ \beta\, c0 \quad (3.12c)$$

$$C_4 = (c3^{**}\ \beta\, c0^{*})\ \gamma\, c1 \quad (3.12d)$$

A quantidade de sub-funções para expressar a função depende da quantidade de valores que ela fornece, exatamente como ocorre para funções de uma entrada:

Valores	*c0*	*c1*	*c2*	*c3*	Sub-funções	Formas	Redução
0 ou (0,1)	Sim				1	C_1	Binário
1 ou (1,2)		Sim			1	C_2	Binário
2 ou (2,3)			Sim		1	C_3	Binário
3 ou (3,0)				Sim	1	C_4	Binário
(0,1,2)	Sim	Sim			2	C_1	Ternário
(1,2,3)		Sim	Sim		2	C_2	Ternário
(2,3,0)			Sim	Sim	2	C_3	Ternário
(3,0,1)	Sim			Sim	2	C_4	Ternário
(0,2)	Sim	Sim	Sim	Sim	2	C_1, C_3	Ternário
(1,3)	Sim	Sim	Sim	Sim	2	C_2, C_4	Ternário
0,1,2,3	Sim	Sim	Sim	Sim	3	C_1, C_2, C_3, C_4	Quaternário

Tabela 3.90

São, ao todo, 4^{16} = 4.294.967.296 funções de quatro valores e duas entradas.

São, ao todo, 2^{16} = 65.536 funções binárias que podem ser utilizadas como sub-funções.

A escolha de qual modo utilizar (C_1, C_2, C_3, C_4) deve ser feita após a realização das quatro análises, pois é difícil prever qual deles sofrerá a maior redução devido às simplificações por linha e coluna completa.

Exemplo:

B \ A (C)	**0**	**1**	**2**	**3**
0	0	1	0	1
1	2	3	2	3
2	0	1	0	1
3	2	3	2	3

Tabela 3.91

B \ A (*c0*)	**0**	**1**	**2**	**3**
0	0	1	0	1
1	0	0	0	0
2	0	1	0	1
3	0	0	0	0

B \ A (*c1*)	**0**	**1**	**2**	**3**
0	1	1	1	1
1	2	1	2	1
2	1	1	1	1
3	2	1	2	1

B \ A (*c2*)	**0**	**1**	**2**	**3**
0	2	2	2	2
1	2	3	2	3
2	2	2	2	2
3	2	3	2	3

B \ A (*c3*)	**0**	**1**	**2**	**3**
0	0	3	0	3
1	3	3	3	3
2	0	3	0	3
3	3	3	3	3

Tabela 3.92

Todas as sub-funções dispensam a conversão para binário, pois possuem pares de linhas e de colunas intercaladas.

$$c0 = A\alpha\bar{\bar{A}}\alpha\bar{B}\alpha\underline{B}$$

$$c1 = A\beta\bar{\bar{A}}\beta\bar{B}\beta\underline{B}$$

$$c2 = A\gamma\bar{\bar{A}}\gamma\bar{B}\gamma\underline{B}$$

$$c3 = A\delta\bar{\bar{A}}\delta\bar{B}\delta\underline{B}$$

$$C_1 = (c0\gamma c1)\delta c2$$

$$C_2 = (c1\delta c2)\alpha c3$$

$$C_3 = (c2\alpha c3)\beta c0$$

$$C_4 = (c3\beta c0)\gamma c1$$

$$C_1 = ((A\alpha\bar{\bar{A}}\alpha\bar{B}\alpha\underline{B})\gamma(A\beta\bar{\bar{A}}\beta\bar{B}\beta\underline{B}))\delta(A\gamma\bar{\bar{A}}\gamma\bar{B}\gamma\underline{B})$$

$$C_2 = ((A\beta\bar{\bar{A}}\beta\bar{B}\beta\underline{B})\delta(A\gamma\bar{\bar{A}}\gamma\bar{B}\gamma\underline{B}))\alpha(A\delta\bar{\bar{A}}\delta\bar{B}\delta\underline{B})$$

$$C_3 = ((A\gamma\bar{\bar{A}}\gamma\bar{B}\gamma\underline{B})\alpha(A\delta\bar{\bar{A}}\delta\bar{B}\delta\underline{B}))\beta(A\alpha\bar{\bar{A}}\alpha\bar{B}\alpha\underline{B})$$

$$C_4 = ((A\delta\bar{\bar{A}}\delta\bar{B}\delta\underline{B})\beta(A\alpha\bar{\bar{A}}\alpha\bar{B}\alpha\underline{B}))\gamma(A\beta\bar{\bar{A}}\beta\bar{B}\beta\underline{B})$$

Solução utilizando irrelevantes:

		A			
	c0*	**0**	**1**	**2**	**3**
	0	0	1	0	1
B	**1**	0	0	0	0
	2	0	1	0	1
	3	0	0	0	0

		A			
	c1*	**0**	**1**	**2**	**3**
	0	1	1	1	1
B	**1**	2	2	2	2
	2	1	1	1	1
	3	2	2	2	2

		A			
	c2*	**0**	**1**	**2**	**3**
	0	2	2	2	2
B	**1**	2	3	2	3
	2	2	2	2	2
	3	2	3	2	3

		A			
	c3*	**0**	**1**	**2**	**3**
	0	0	0	0	0
B	**1**	3	3	3	3
	2	0	0	0	0
	3	3	3	3	3

		A			
	c0**	**0**	**1**	**2**	**3**
	0	0	1	0	1
B	**1**	0	1	0	1
	2	0	1	0	1
	3	0	1	0	1

		A			
	c1**	**0**	**1**	**2**	**3**
	0	2	1	2	1
B	**1**	2	1	2	1
	2	2	1	2	1
	3	2	1	2	1

		A			
	c2**	**0**	**1**	**2**	**3**
	0	2	3	2	3
B	**1**	2	3	2	3
	2	2	3	2	3
	3	2	3	2	3

		A			
	c3**	**0**	**1**	**2**	**3**
	0	0	3	0	3
B	**1**	0	3	0	3
	2	0	3	0	3
	3	0	3	0	3

Tabela 3.93

Algumas das funções para a segunda sub-função coincidem, neste exemplo, com as funções para a segunda sub-função, mostradas na tabela anterior.

$$c0^{**} = A\alpha\overline{\overline{A}} \qquad c0^{*} = A\alpha\overline{\overline{A}}\alpha\overline{B}\alpha\underline{B}$$

$$c1^{**} = A\beta\overline{\overline{A}} \qquad c1^{*} = \overline{B}\beta\underline{B}$$

$$c2^{**} = A\gamma\overline{\overline{A}} \qquad c2^{*} = A\gamma\overline{\overline{A}}\gamma B\gamma\overline{\overline{B}}$$

$$c3^{**} = A\delta\overline{\overline{A}} \qquad c3^{*} = B\delta\overline{\overline{B}}$$

$$C_1 = (c0^{**}\gamma c1^{*})\delta c2 \qquad C_1 = ((A\alpha\overline{\overline{A}})\gamma(\overline{B}\beta\underline{B}))\delta(A\gamma\overline{\overline{A}}\gamma\overline{B}\gamma\underline{B})$$

$$C_2 = (c1^{**}\delta c2^{*})\alpha c3 \qquad C_2 = ((A\beta\overline{\overline{A}})\delta(A\gamma\overline{\overline{A}}\gamma B\gamma\overline{\overline{B}}))\alpha(A\delta\overline{\overline{A}}\delta\overline{B}\delta\underline{B})$$

$$C_3 = (c2^{**}\alpha c3^{*})\beta c0 \qquad C_3 = ((A\gamma\overline{\overline{A}})\alpha(B\delta\overline{\overline{B}}))\beta(A\alpha\overline{\overline{A}}\alpha\overline{B}\alpha\underline{B})$$

$$C_4 = (c3^{**}\beta c0^{*})\gamma c1 \qquad C_4 = ((A\delta\overline{\overline{A}})\beta(A\alpha\overline{\overline{A}}\alpha\overline{B}\alpha\underline{B}))\gamma(A\beta\overline{\overline{A}}\beta\overline{B}\beta\underline{B})$$

Capítulo 4
O aplicativo

4.1 Aspectos de Desenvolvimento

O software utilizado no desenvolvimento e na compilação do aplicativo foi o Microsoft Visual Basic 6.0 Enterprise Edition.

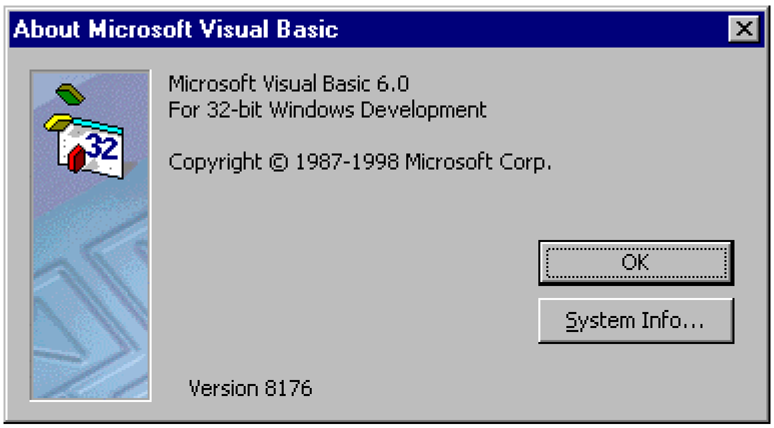
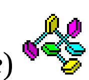


Figura 4.1 – O software utilizado.

Os arquivos utilizados pelo compilador são mostrados abaixo. Cada arquivo do tipo “FRM” (*Visual Basic Form File*) consite em um “*form*” , nome dado pelos aplicativos visuais às janelas. Os arquivos do tipo “BAS” (*Visual Basic Module*)  consistem em “*module*”, conjuntos de procedimentos desvinculados a qualquer *form*. O arquivo principal, que gerencia todos os demais arquivos, é chamado de “*Project*” (*Visual Basic Project*) “VBP”.

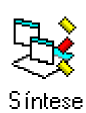


Figura 4.2

Figura 4.3

Abaixo, está a lista dos arquivos que constituem o código fonte do projeto.

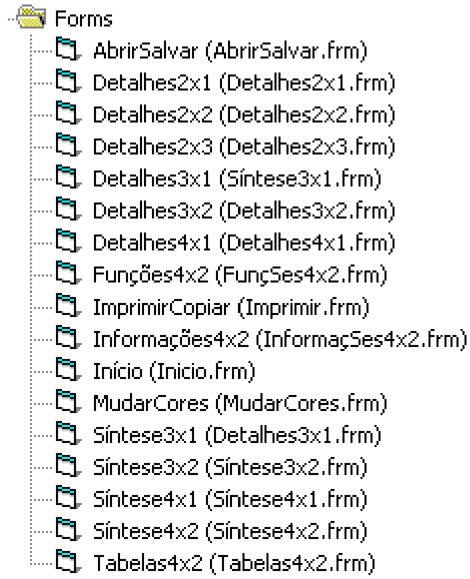


Figura 4.4 – *Forms.*

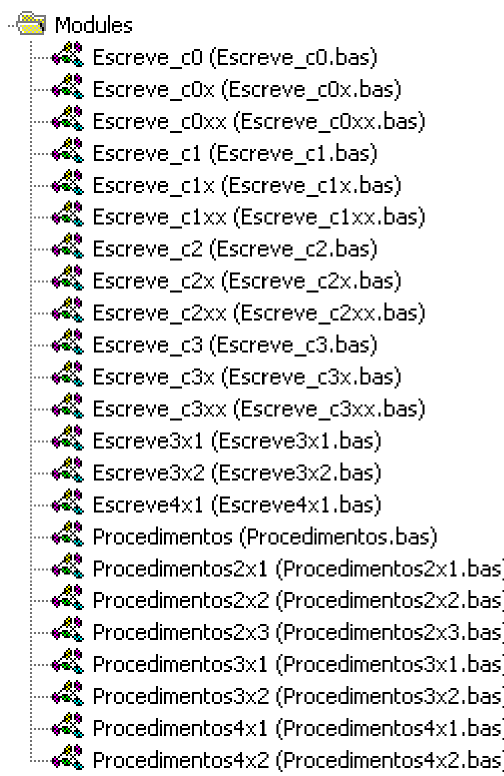


Figura 4.5 – *Modules.*

4.2 Apresentação do Software

O programa exibe uma janela por vez, não havendo cascateamento. O objetivo disso é permitir um melhor aproveitamento da área útil da tela, permitindo a visualização de outros softwares ao mesmo tempo. Por esse motivo, cada janela apresenta as informações na maneira mais condensada possível. Abaixo, é mostrado o logotipo do software.

Figura 4.6 – Logotipo.

Ao iniciar a execução do programa, uma janela de apresentação é exibida, mostrando informações a respeito do produto. É fornecido o endereço de correio eletrônico do autor do software, a versão, a data da última atualização e o logotipo.

Figura 4.7 – Janela de apresentação.

Abaixo, são mostrados os ícones utilizados pelo aplicativo: (16x16x16, 32x32x16, 32x32x256, 48x48x16, 48x48x256) – largura x altura x cores.

Figura 4.8 – Ícones.

Ao realizar uma síntese, o usuário deve escolher dois parâmetros: o número de entradas (ou variáveis) e o número de valores. Para expressar todas as combinações, utiliza-se o padrão “valores” X “variáveis”.

Sendo assim, as possibilidades são: 2x1, 2x2, 2x3, 3x1, 3x2, 4x1, 4x2. As opções para dois valores foram disponibilizadas para que se veja que o método utilizado independe da quantidade de valores. No entanto, este método é realmente útil para três e quatro valores. Todas as janelas cabem na tela no menor modo possível, 640x480 pixels.

O menu de opções é mostrado abaixo:

Figura 4.9 – Menus.

- Arquivo
  - Abrir: Abre um arquivo salvo anteriormente.
  - Salvar: Grava a síntese em um arquivo em disco.
  - Copiar: Copia a síntese para a área de transferência (*clipboard*).
  - Imprimir: Envia a síntese para a impressora.
  - Sair: Sai do programa.
- Visualizar (disponível apenas para três e quatro valores).
  - Normal: Mostra apenas a tabela verdade, a função minimizada e os comandos de manipulação. A janela no modo normal recebe o nome (caption) de “Síntese valores X entradas”. Os três modos a seguir só são exibidos separadamente no modo 4x2. Nos demais modos, são exibidos sob o nome de “Detalhes”.
  - Tabelas: Mostra as tabelas binárias utilizadas na criação de tabelas MVL.
  - Funções: Mostra as sub-funções binárias utilizadas na criação de funções MVL e as fórmulas para as funções completas.
  - Informações: Mostra quantos termos cada tabela verdade binária possui e as fórmulas para as funções completas.
- Tipos: Seleciona os tipos 2x1, 2x2, 2x3, 3x1, 3x2, 4x1, 4x2.
- Cores
  - Sem: A cor de fundo utilizada é “*Window Background*” (&H80000005&).
  - Padrão 1: Seleciona o padrão 1.
  - Padrão 2: Seleciona o padrão 2.
  - Usuário: Seleciona as cores definidas pelo usuário.
  - Definir: Define as cores do usuário.

No modo 4x2, os detalhes foram divididos em tabelas, funções e informações para que coubessem numa tela com tamanho de 640x480 pixels.

Este produto executa a tarefa inversa daquela realizada pelo ELOmv (Equações Lógicas Multi-Valores), , de autoria de Luciana Prado do Nascimento [09].

ELOmv recebe a função na forma algébrica e gera a tabela-verdade equivalente. Este produto recebe a tabela-verdade e gera a função na forma algébrica.

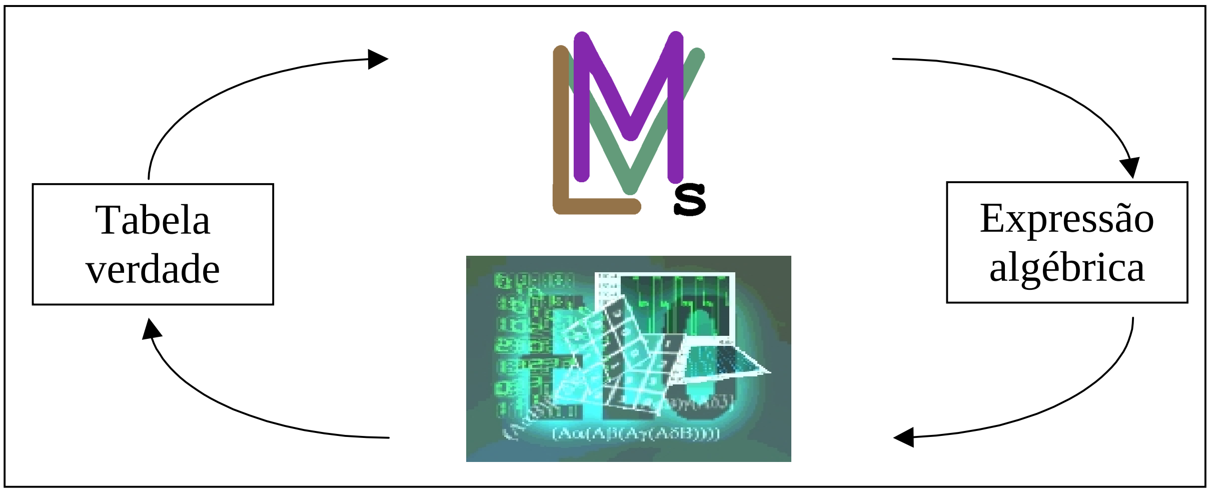


Figura 4.10

O emprego destes dois aplicativos em conjunto possibilita, ao usuário, usufruir de total disponibilidade para fazer as manipulações desejadas em funções MVL.

Há boas perspectivas para uma futura integração nestes programas, de modo que a conversão seja realizada nos dois sentidos no mesmo aplicativo.

Ambas as conversões podem ser muito trabalhosas para de fazer manualmente, especialmente a conversão da forma algébrica para a tabela, pois a expressão pode ser muito grande. Em se tratando de trabalho manual, a conversão da tabela para a expressão normalmente é mais rápida do que a conversão inversa.

No entanto, em se tratando de automação, a conversão da tabela para a expressão é mais trabalhosa, pois envolve tomadas de decisões, o que não acontece na conversão contrária. A obtenção da expressão sem minimização é tarefa simples, mas a minimização envolve "raciocínio" por parte do computador, e isso exige um trabalho de programação oneroso.

Em virtude do fato de que este programa manipula arquivos do tipo texto, existe a necessidade de realizar uma mudança na maneira como a expressão é escrita, pois, no modo texto, não é possível utilizar os caracteres gregos ALFA, BETA, GAMA e DELTA nem colocar as barras para os deslocadores TOPO, DUPLO TOPO e BASE.

O motivo de se usar o modo texto é o de desfrutar de sua universalidade, podendo ser exibido, impresso e manipulado em qualquer ambiente, até mesmo em outros sistemas operacionais. O resultado da síntese pode ser manipulado de todas essas maneiras.

O ELOmv também faz uso do modo texto, mas utiliza um outro padrão de grafia.

expressão	L.M.V.S.	ELOmv
$A\alpha B$	A a B	A * B
$A\beta B$	A b B	A + B
$A\gamma B$	A c B	A – B
$A\delta B$	A d B	A / B
$\overline{A}$	/A	At
$\overline{\overline{A}}$	//A	Ad
$\underline{A}$	A/	Ab

Tabela 4.1 – Padrões de grafia utilizados.

O usuário precisa ter consciência destas diferenças para poder utilizar os dois softwares em conjunto, bem como para escrever as expressões utilizando os caracteres especiais.

A função é fornecida pelo usuário na forma de tabela-verdade. O número de dimensões da matriz de entrada equivale à quantidade de entradas (ou de variáveis). A palavra "entrada" é utilizada em referência às portas lógicas; a palavra "variável" é mais genérica e refere-se apenas à análise algébrica.

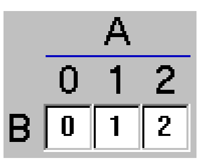
Para duas entradas, foi feita a escolha (arbitrária) de se utilizar as colunas (eixo *X*) para a variável *A* e as linhas (eixo *Y*) para a variável *B* (página 31).

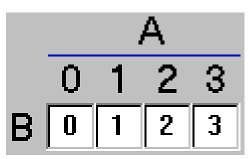
O nome da variável de saída depende da quantidade de entradas:

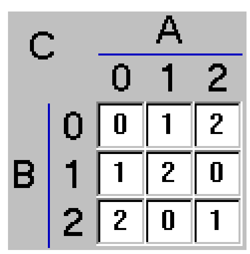
Entradas	Saída	Entrada(s)
1	*B=B(A)*	*A*
2	*C=C(A,B)*	*A,B*
3	*D=D(A,B,C)*	*A,B,C*

Tabela 4.2

Abaixo, são mostrados exemplos para as matrizes de entrada para 3x1, 4x1, 3x2, 4x2, respectivamente.







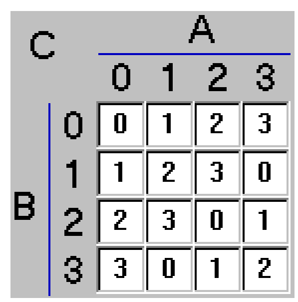


Figura 4.11 **Figura 4.12** **Figura 4.13** **Figura 4.14**

A matriz de entrada é composta de um conjunto de caixas de texto (objeto *TextBox*). A síntese é feita à medida que a matriz é digitada, em tempo real. Se for digitado um número acima do limite para a quantidade de valores lógicos escolhida, ele é automaticamente convertido para zero.

A função sintetizada é exibida também em uma caixa de texto. No ambiente visual, a caixa de texto possui uma propriedade (*Property*) chamada *Locked*, para a qual um valor *False* (falso, 0) permite a edição de conteúdo em tempo de execução (*run time*), e um valor *True* (verdadeiro, 1) inibe esta edição. Há, também, outra propriedade, chamada *Apearance* (aparência), que pode ser 0 – *Flat* (plano) ou 1 – *3D*.

	Locked	*Apearance*
Matriz (entrada)	0	1
Função (saída)	1	0

Tabela 4.3

4.3 Modos de exibição

Será mostrada, agora, a maneira como as janelas são exibidas, utilizando alguns exemplos. A opção sem cores foi a escolhida durante a captura das imagens.

Figura 4.15 – 2x1.

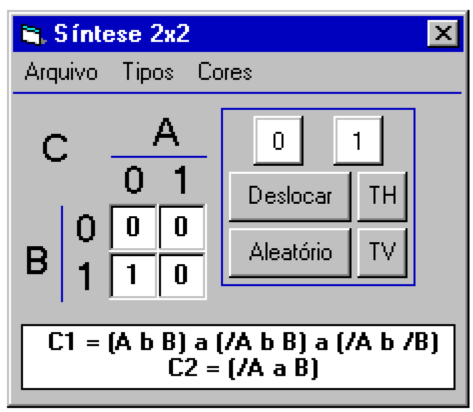


Figura 4.16 – 2x2.

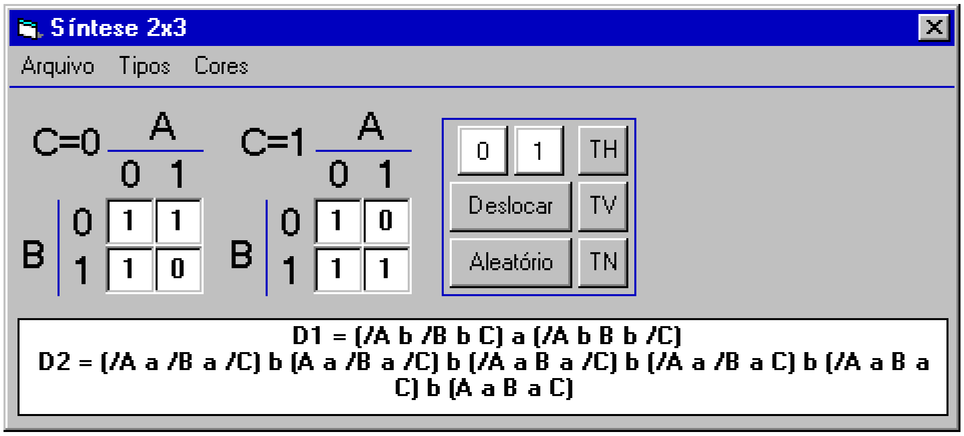


Figura 4.17 – 2x3.

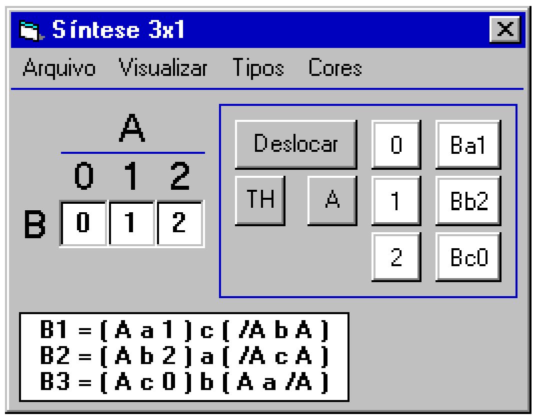


Figura 4.18 – 3x1.

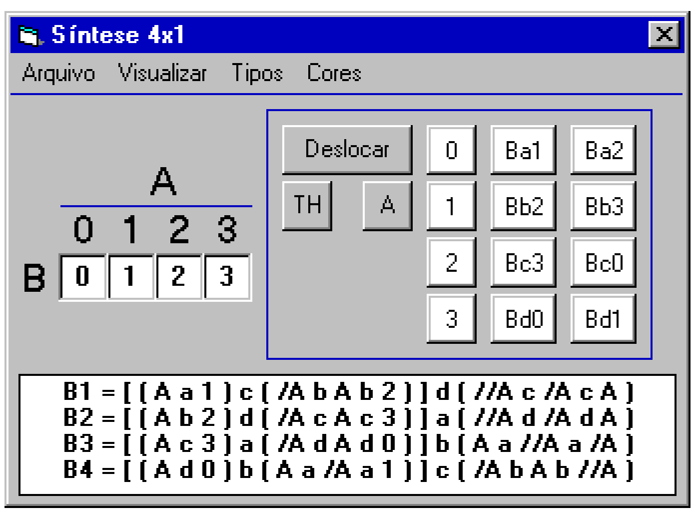


Figura 4.19 – 4x1.

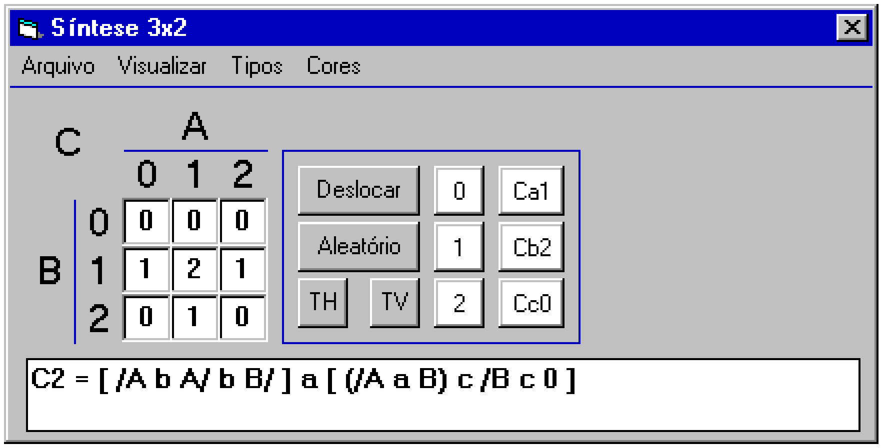


Figura 4.20 – 3x2.

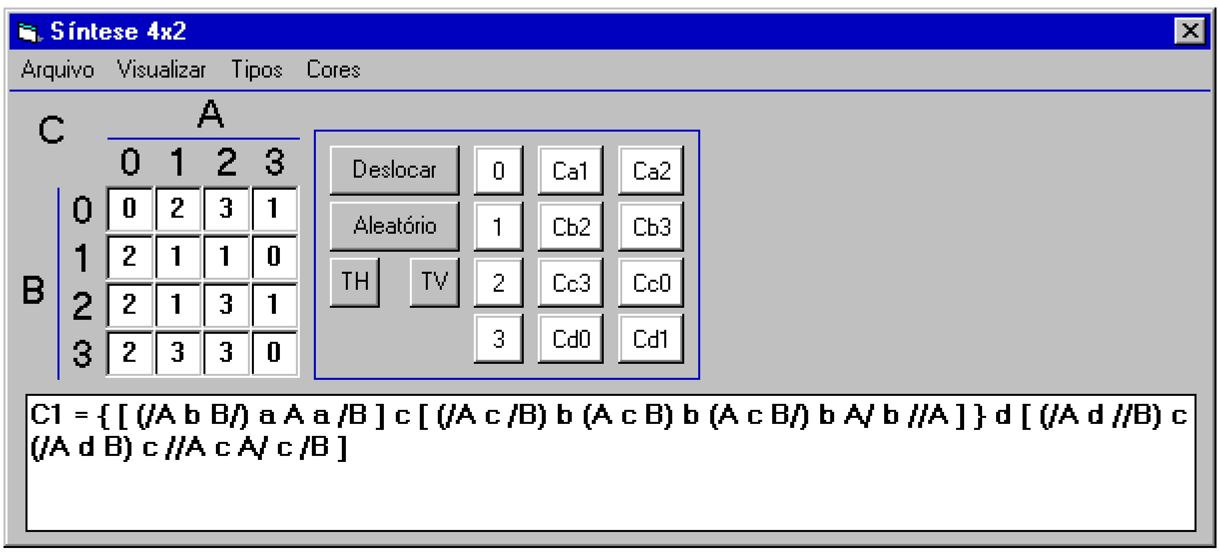


Figura 4.21 – 4x2.

4.4 Janelas com Detalhes

A seguir, são mostradas as janelas no modo de exibição com detalhes.

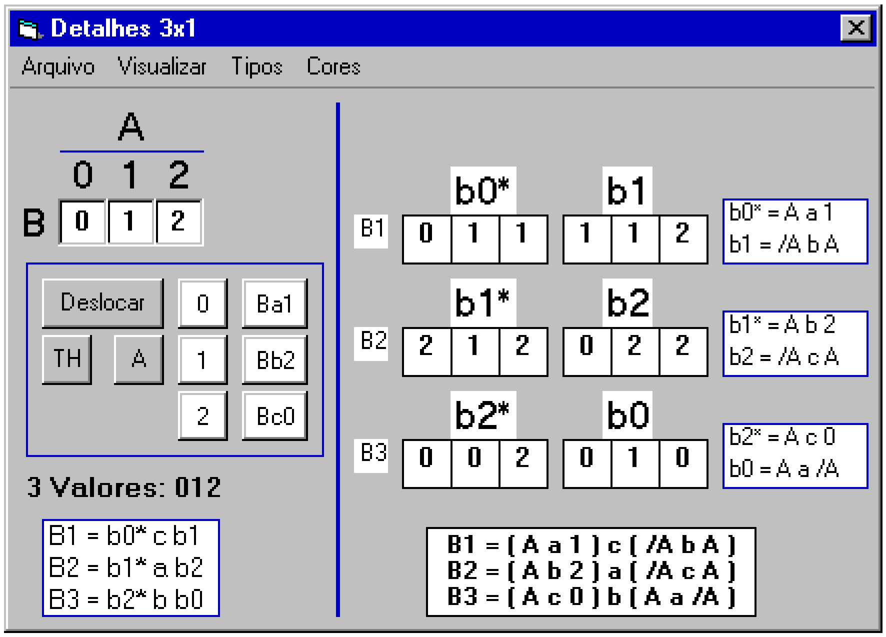


Figura 4.22 – 3x1.

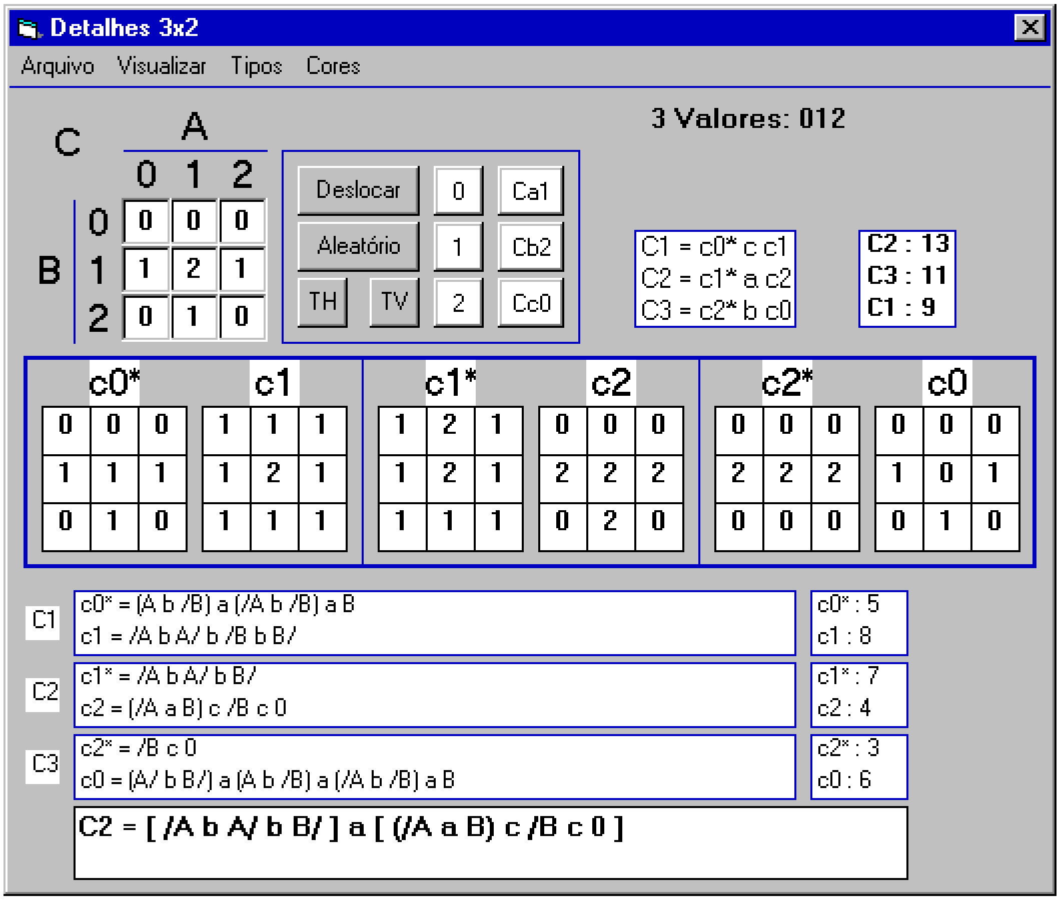


Figura 4.23 – 3x2.

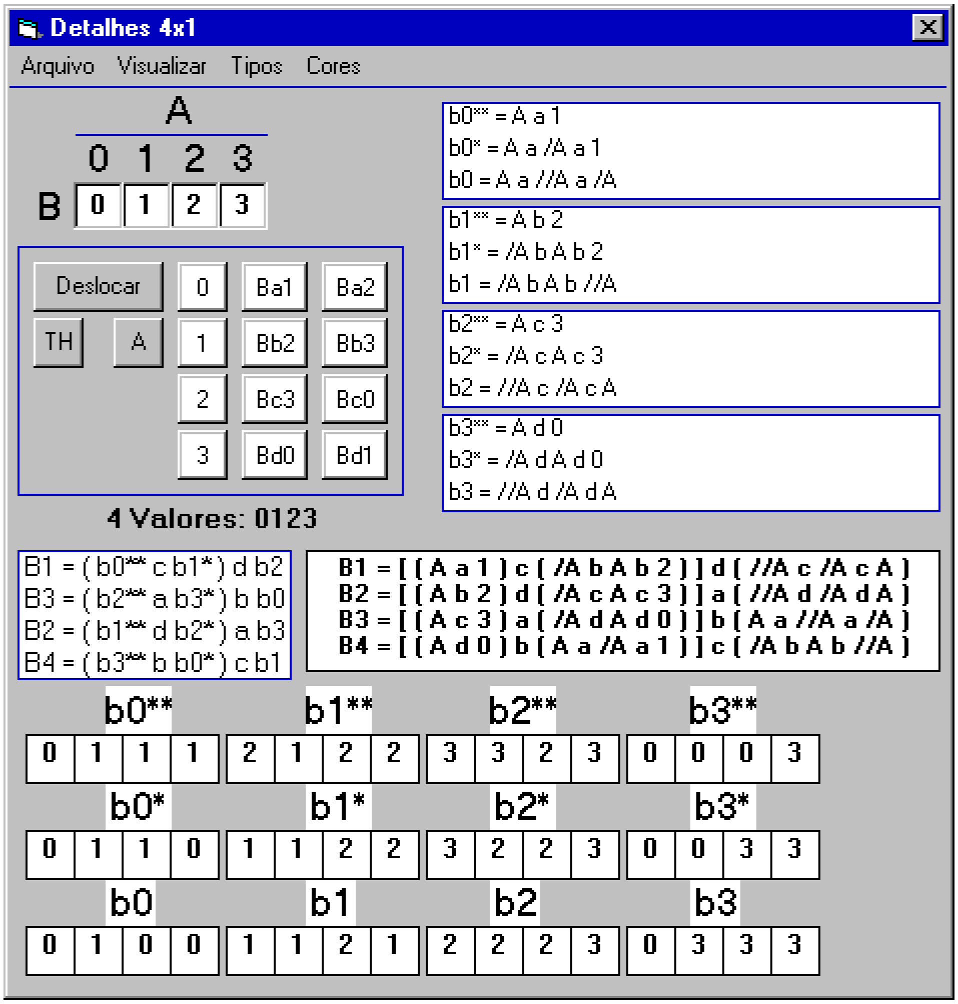


Figura 4.24 – 4x1.

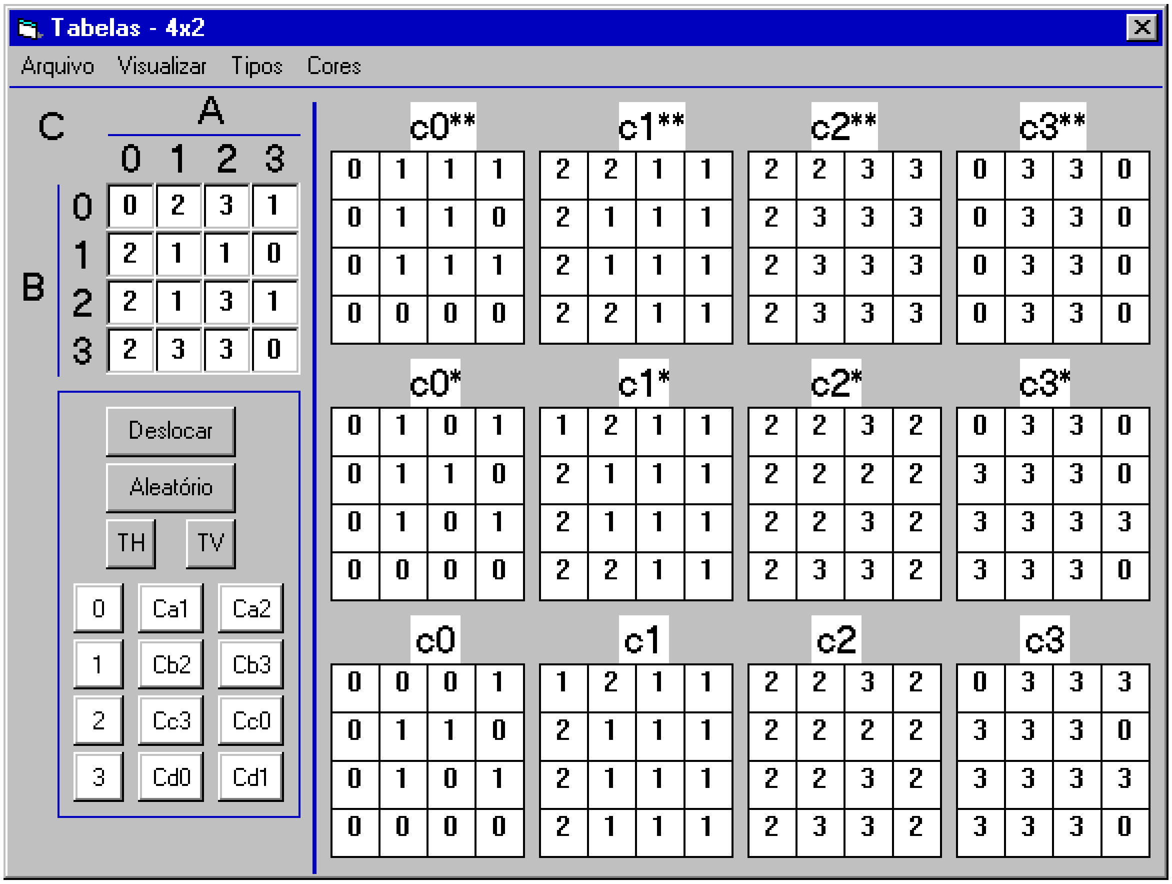


Figura 4.25 – 4x2.

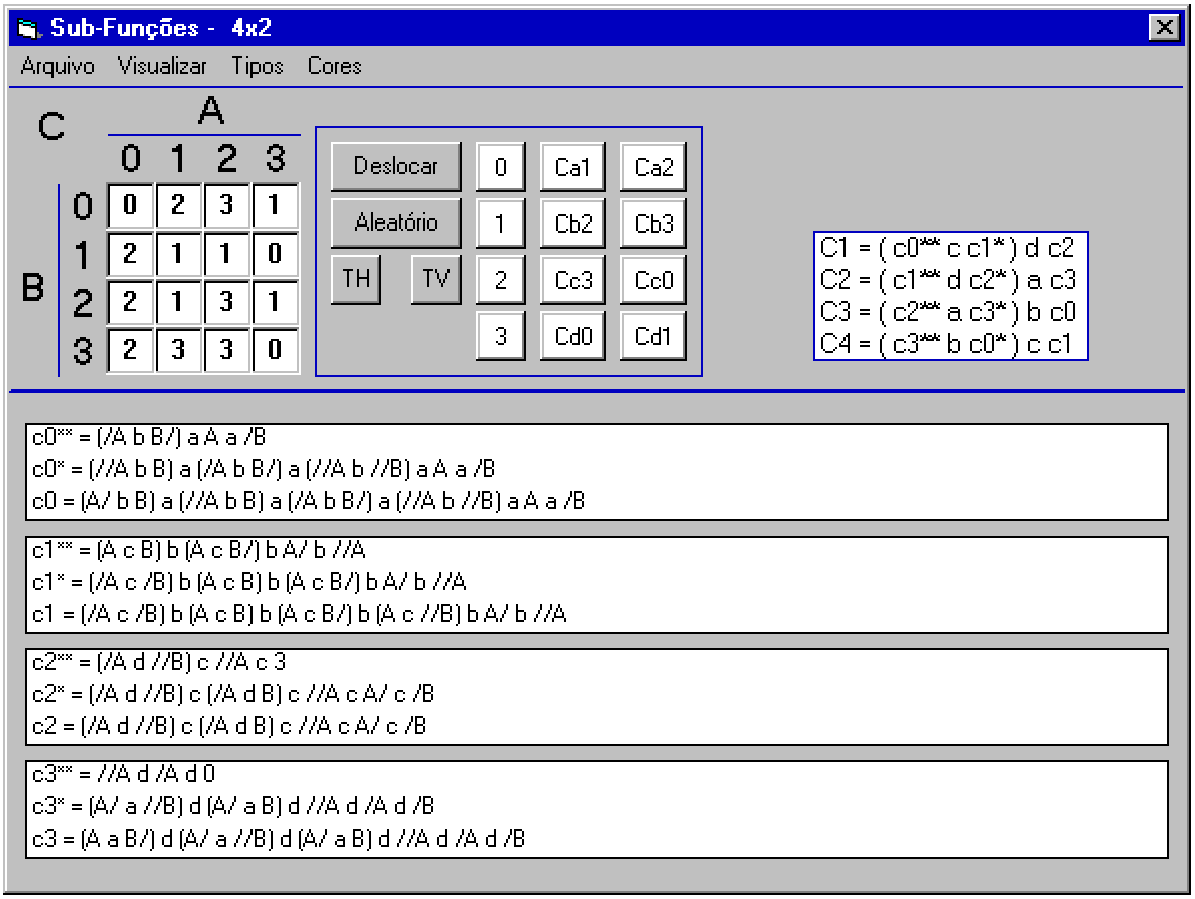


Figura 4.26 – 4x2.

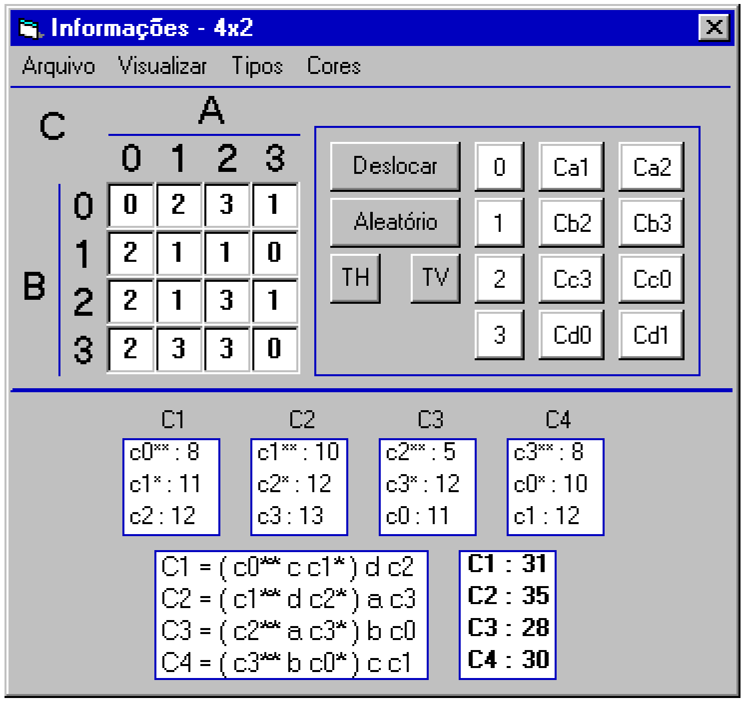


Figura 4.27 – 4x2.

4.5 Comandos

Os comandos de manipulação (objeto *CommandButton*) existem para facilitar a digitação da tabela e são mostrados na figura abaixo e explicados logo a seguir.

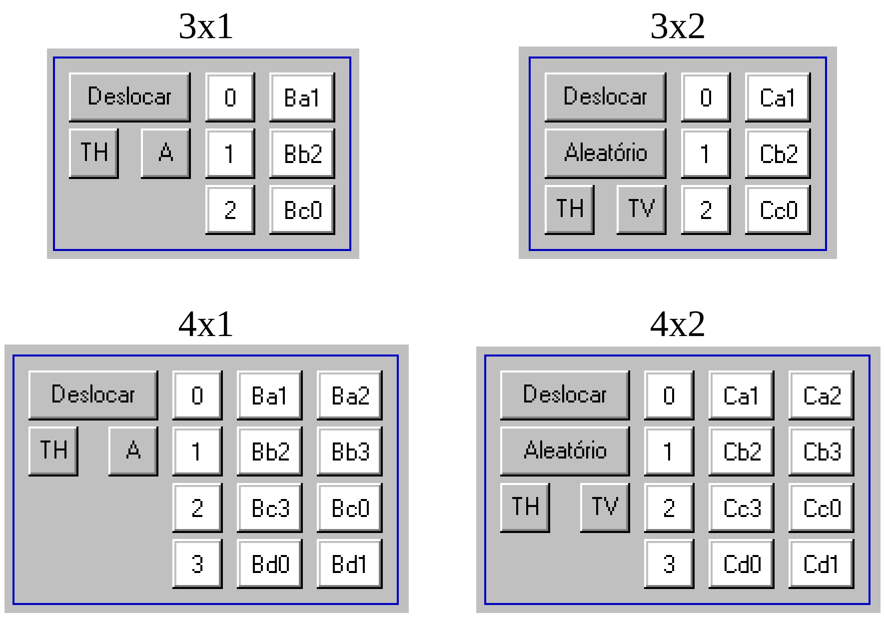


Figura 4.28

- 0,1,2,3 : Preenche toda a tabela (todas as combinações de entrada) com o valor especificado. É útil quando na digitação quando um determinado valor predominar.
- (Ba1, Bb2, Bc0), (Ca1, Cb2, Cc0), (Ba1, Bb2, Bc3, Bd0), (Ca1, Cb2, Cc3, Cd0): Executa a conversão para binário. Quando a função fornecer apenas dois valores, estes comandos são desabilitados. Obviamente, estes comandos não estão disponíveis para lógica de dois valores.
- (Ba2, Bb3, Bc0, Bd1), (Ca2, Cb3, Cc0, Cd1) : Executa a conversão para ternário. Disponível apenas para lógica de quatro valores. Quando a função fornecer apenas três valores, estes comandos são desabilitados.

- TH, TV : Translação horizontal (para a esquerda) e translação vertical (para cima), respectivamente. Em outras palavras, TH desloca a variável “A” para a esquerda e TV desloca a variável “B” para cima. TV só está disponível para duas ou três variáveis. Para 2x3, há também o comando TN - Translação normal, perpendicular ao plano da tela, o deslocamento da terceira variável “C”.
- Deslocar : Realiza uma operação TOPO na matriz de entrada.
- Aleatório : Cria uma matriz aleatória, usando todos os valores possíveis. Disponível apenas para duas ou três entradas.
- A : Disponível apenas para uma entrada (3x1, 4x1). Este botão coloca a saída *B=B(A)* como uma imagem da entrada *A*.

Os botões 0 1 2 3 e os da conversão para binário e ternário são exibidos em branco (propriedade *BackColor*) na figura 4.28 porque a opção de cores utilizada foi a “sem cores” e a cor de fundo do Windows™ (*window background*) para janela é a branca. No entanto, se a cor de fundo do Windows™ escolhida pelo usuário for outra, tais botões assumirão aquela cor.

4.6 Quantidade de termos

A princípio, no que se refere à síntese de funções, a grandeza “quantidade de termos” não tem nenhuma aplicação prática, no entanto, ela é fornecida mesmo assim, pois, em se tratando de implementação de portas lógicas em circuitos integrados, alguns valores podem exigir maior consumo de potência no dispositivo, devendo, por isso, ser evitados. A função pode ser obtida de diversas maneiras; deve-se, por esse motivo, privilegiar aquela que utilize aquele valor o mínimo possível. Ainda não se sabe se esta análise é relevante diante de outros aspectos da elaboração de circuitos, mas, mesmo assim, ela pode ter outras utilidades que ainda são desconhecidas.

Quando se fala em “termos”, refere-se às combinações de entrada das sub-funções binárias para as quais a saída corresponde ao valor dominante para a operação utilizada (ver página 31).

Supondo o seguinte exemplo:

C		A			
		0	1	2	3
B	0	1	3	3	1
	1	1	0	0	2
	2	2	0	1	3
	3	1	1	1	3

Figura 4.29 – Matriz de entrada para o exemplo.

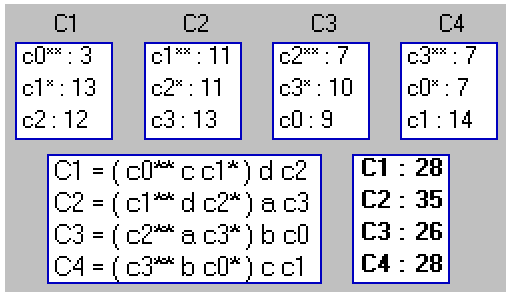


Figura 4.30 – Quantidades de termos para o exemplo.

A quantidade de termos para cada uma das doze sub-funções pode ser obtida pelas tabelas da figura abaixo. Por exemplo, para c0**, basta contar a quantidade de ocorrências do valor zero, que é três.

A quantidade de termos para C1, por exemplo, será de:

c0** + c1* + c2, 3+13+12=28.

c0**

1	1	1	1
1	0	0	1
1	0	1	1
1	1	1	1

c1**

1	1	1	1
1	2	1	2
2	2	1	2
1	1	1	1

c2**

2	3	3	3
2	2	2	2
2	3	3	3
2	3	3	3

c3**

3	3	3	3
0	0	0	3
0	0	0	3
0	0	0	3

c0*

1	0	0	1
1	0	0	1
1	0	1	0
1	1	1	0

c1*

1	1	1	1
1	1	1	2
2	1	1	2
1	1	1	1

c2*

2	3	3	2
2	2	2	2
2	3	2	3
2	2	2	3

c3*

3	3	3	3
3	0	0	3
3	0	0	3
3	0	0	3

c0

1	0	0	1
1	0	0	0
0	0	1	0
1	1	1	0

c1

1	1	1	1
1	1	1	2
2	1	1	1
1	1	1	1

c2

2	3	3	2
2	2	2	2
2	2	2	3
2	2	2	3

c3

3	3	3	3
3	0	0	3
3	0	3	3
3	3	3	3

Figura 4.31 – Tabelas das sub-funções binárias para o exemplo.

4.7 Técnicas de minimização

Várias técnicas são empregadas na obtenção da forma mínima. Elas são citadas na mesma ordem em que são empregadas.

4.7.1- Uso de irrelevantes

Nas conversões para binário que transformam a tabela MVL fornecida em várias tabelas binárias, os valores diferentes do dominante e do secundário são substituídos pelo dominante; no entanto, cada ocorrência do dominante acarreta no aparecimento de um termo na sub-expressão binária.

O uso de irrelevantes (X) é uma poderosa arma para a eliminação de termos na expressão, pois elimina tais termos diretamente nas tabelas das sub-funções binárias. Esse artifício consiste em fazer a substituição pelo secundário (e não pelo dominante) quando o valor for irrelevante, para diminuir o tamanho da expressão.

O uso desta técnica torna o trabalho manual muito mais extenso quando se deseja obter todas as formas da expressão (C_1, C_2, C_3, C_4, por exemplo), pois, para este caso, ao invés de se ter apenas quatro sub-funções binárias (c0,c1,c2,c3), tem-se doze (c0**,c1**,c2**,c3**, c0*,c1*,c2*,c3*, c0,c1,c2,c3) mas, para uma ferramenta automatizada, isso não é problema.

	Sem irrelevantes	Com irrelevantes
C1	c0 c c1	c0* c c1
C2	c1 d c2	c1* d c2
C3	c2 a c0	c2* a c0

Tabela 4.4 – Três valores.

	Sem irrelevantes	Com irrelevantes
C1	(c0 c c1) d c2	(c0** c c1*) d c2
C2	(c1 d c2) a c3	(c1** d c2*) a c3
C3	(c2 a c3) b c0	(c2** a c3*) b c0
C4	(c3 b c0) c c1	(c3** b c0*) c c1

Tabela 4.5 – Quatro valores.

O uso de irrelevantes deve ser utilizado com cautela. Se todos eles forem substituídos pelo secundário, pode-se obter uma expressão não minimizada. Para resolver esse problema, utiliza-se a segunda técnica.

4.7.2- Formação de linhas e colunas completas

Os irrelevantes são substituídos pelo dominante quando existe a possibilidade de formar uma linha ou coluna completa. Se esta possibilidade não existir devido à presença de uma ocorrência do valor secundário na linha ou coluna analisada, o irrelevante é substituído pelo secundário.

Quando a linha ou coluna não possui o dominante, os irrelevantes são substituídos pelo secundário, mesmo que seja possível completar a linha ou coluna, pois, nesse caso, isso só iria aumentar o tamanho da expressão.

Os dominantes gerados por meio da substituição de um irrelevante devem ser ignorados nesta análise, ou seja, devem ser considerados como irrelevantes até que a análise seja concluída. Também devem ser ignorados os dominantes que já foram empregados na formação de uma linha ou coluna completa, para evitar a formação de uma linha ou coluna na qual todos os dominantes originais já façam parte de outras linhas ou colunas completas.

Por esses motivos, há muitas maneiras de se fazer a análise, pois, pode fazer diferença começar pelas linhas ou pelas colunas; também pode fazer diferença a escolha da linha ou da coluna por onde começar a análise. Sendo assim, tem-se $6! = 720$ possibilidades de análise para 3x2 (3 linhas + 3 colunas = 6) e $8! = 40320$ para 4x2. Verificou-se, por experimentação, que a substituição dos irrelevantes é realizada da melhor forma possível em quase todos os casos; no entanto, tem-se 2^9 sub-funções binárias para três valores, para quatro valores, esse valor sobe para 2^{16}, ou seja, é impossível analisar caso a caso. É justamente por esse motivo que as tabelas das sub-funções binárias são expostas, para que o usuário possa verificar se concorda ou não com as substituições realizadas.

Apesar de fácil de ser realizada por uma pessoa, esta técnica, que envolve intuição, consiste no algoritmo do programa cuja criação consumiu mais tempo. É a única etapa da execução do programa na qual o computador precisa “pensar” como um ser humano.

Esta análise só é realizada para duas entradas. Para uma entrada, todos os irrelevantes são substituídos pelo secundário, exceto para funções quaternárias que forneçam apenas dois valores não visinhos (ver exemplo na página 63).

4.7.3- Eliminação da conversão para binário

Toda sub-função binária proveniente de uma função MVL necessita, a princípio, da conversão para binário, para eliminar possíveis ocorrências de valores espúrios (nem verdadeiro - dominante, nem falso - secundário). No entanto, para alguns casos, essa conversão não é necessária, pois não há ocorrências de valores espúrios.

Para três valores, basta que duas ocorrências do dominante estejam dispostas em diagonal para que a conversão para binário seja desnecessária (página 48).

Para quatro valores, três ocorrências do valor dominante em diagonal também tornam a conversão para binário dispensável. Quatro ocorrências do dominante formando um quadrado 3x3 (cada dominante numa quina) também possibilitam essa eliminação da conversão para binário (ver exemplos na página 68).

O programa deve fazer essa verificação e descobrir se a conversão para binário pode ou não ser eliminada. Esta análise deve ser feita em último lugar, pois, uma vez que quanto maior for o número de ocorrências do valor dominante nas sub-funções binárias menor será a chance de precisar da conversão para binário, o uso de irrelevantes aumenta a probabilidade desta conversão ser necessária.

4.7.4- Escolha da menor forma

Para lógica de dois valores, tem-se duas formas de expressão da função, C0 (disjuntiva) e C1 (conjuntiva). Igualmente, para três e quatro valores, temos, respectivamente, três e quatro formas. A menor delas deve ser escolhida.

Nesta escolha, o software não pode analisar a quantidade de termos, pois, ao se criar linhas e colunas completas, a quantidade de termos aumenta, mas o tamanho da expressão diminui. O critério de escolha é o tamanho da cadeia (*string*) da expressão. Aquela que contiver o menor número de caracteres será a escolhida.

4.8 Menu Arquivo

As janelas para restaurar um arquivo ou gravá-lo são mostradas abaixo. A extensão utilizada é "MVL". Somente arquivos com esta extensão podem ser escolhidos. O diretório ou pasta inicial para procura por arquivos é aquele onde o programa está instalado. Pode-se escolher o arquivo (objeto *FileListBox*), o diretório (objeto *DirListBox*) e a unidade local (objeto *DriveListBox*). A opção para unidades de rede não foi incluída porque o utilitário de desenvolvimento utilizado não fornece tal objeto.

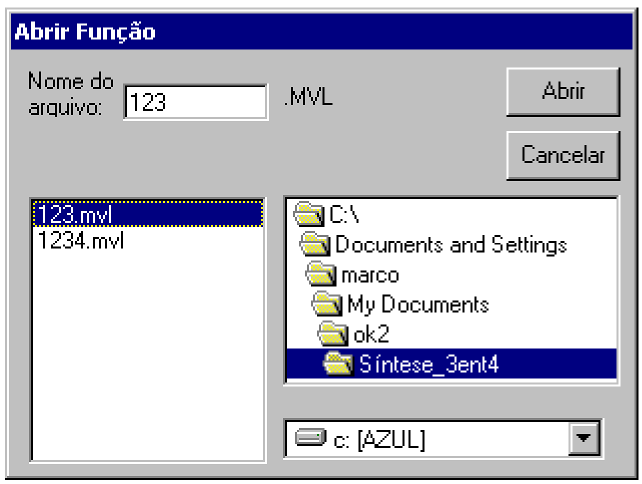


Figura 4.32

Figura 4.33

As janelas usadas para copiar uma função ou imprimí-la são mostradas abaixo. O algoritmo utilizado para gravar, copiar ou imprimir a função é o mesmo. Utiliza-se o modo texto.

Figura 4.34 – Copiar.

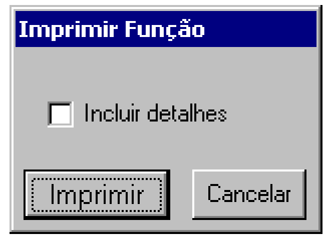


Figura 4.35 – Imprimir.

O algoritmo para a criação do texto permite incluir ou não detalhes. Na opção sem detalhes (*default*), apenas a tabela verdade e a função mínima são mostrados. Na opção com detalhes, são mostradas todas as funções, as tabelas binárias e as sub-funções binárias. Se houver uma entrada e todos os valores forem utilizados, não há forma mínima. Todas as possibilidades são iguais em tamanho e, nesse caso, todas elas são expostas no modo sem detalhes.

Quando o programa é executado pela primeira vez, um arquivo de inicialização (.INI) é criado no diretório onde o programa está armazenado. Tal arquivo contém informações sobre o estado atual da execução do programa.

Figura 4.36

Quando a execução do programa é finalizada, este arquivo conterá todas as informações do programa relativas à ocasião de seu encerramento. Quando o programa for executado novamente, ele partirá exatamente no estado em que se encontrava quando seu uso foi finalizado.

A criação ou atualização do arquivo de inicialização utiliza o mesmo algoritmo utilizado para salvar, copiar ou imprimir, com a exceção de que, neste caso, utiliza-se apenas a opção sem detalhes; as cores definidas pelo usuário também são incluídas, para não se perderem quando se termina a execução do programa.

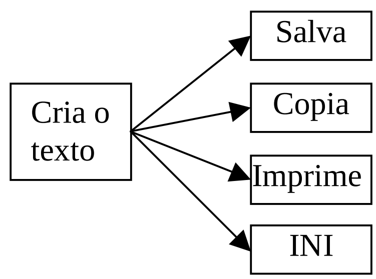


Figura 4.37

4.9 Menu Cores

As cores são utilizadas para melhorar a percepção dos valores por parte do usuário. No entanto, quando o Windows™ é utilizado no modo de cores de 8 bits (256 cores), as cores utilizadas nos padrões 1 e 2 serão mostradas por *dither*, um método de utilização de cores de 24 bits em um modo de 8 bits, que utiliza cores indexadas (*indexed collors*). Isso causará um efeito desagradável ao usuário, sendo melhor, então, não utilizar cores. A opção de uso de cores é recomendada para os modos de 16, 24 e 32 bits.

As cores dos padrões 1 e 2 são mostradas a seguir:

Cor	R	G	B	RGB	Nome	Sigla
1	204	255	204	CCFFCCh	Verde	G
2	255	204	204	CCCCFFh	Vermelho	R
3	204	204	255	FFCCCCh	Azul	B
4	255	255	204	CCFFFFh	Amarelo	Y

Tabela 4.6 – Padrão 1.

Cor	R	G	B	RGB	Nome	Sigla
1	255	255	204	CCFFFF	Amarelo	Y
2	255	204	255	FFCCFF	Rosa	M
3	204	255	255	FFFFCC	Ciano	C
4	204	204	205	FFCCCC	Azul	Y

Tabela 4.7 – Padrão 2.

Para três valores, o padrão 1 corresponde ao modo RGB, as cores primárias, e o padrão 2 corresponde ao modo YMC, as cores secundárias.

O usuário, ao configurar suas próprias cores, deve configurar quatro cores, sendo cada uma delas associada a um valor (0, 1, 2, 3), mesmo em lógica de dois ou três valores. Cada cor é exibida em quatro formas, como mostrado abaixo.

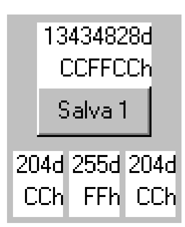


Figura 4.38

As duas formas superiores são as formas implícitas, nas bases 10 e 16, respectivamente, onde se tem um número originalmente na base 256, onde cada dígito (0-255, 1 byte) corresponde a uma cor primária (padrão RGB). As duas formas inferiores são as formas explícitas, nas bases 10 e 16, respectivamente, onde cada dígito (0-255) é mostrado explicitamente. O byte (ou dígito) menos significativo é atribuído ao R (vermelho); o mais significativo é atribuído ao B (azul). A forma implícita é obtida por:

$$cor = R*256^0 + G*256^1 + B*256^2 \quad (4.1)$$

As cores são ajustadas por meio de três barras de rolagem verticais (objeto *VscrollBar*) na área de edição de cores, podendo esta cor ser gravada em qualquer uma das quatro posições. Cinco quadros (objeto *PictureBox*) (um para cada cor e um para a área de edição de cores) são utilizados para exibir as respectivas cores. Clicando nos quadros referentes às quatro posições, a respectiva cor será transferida para a área de edição de cores.

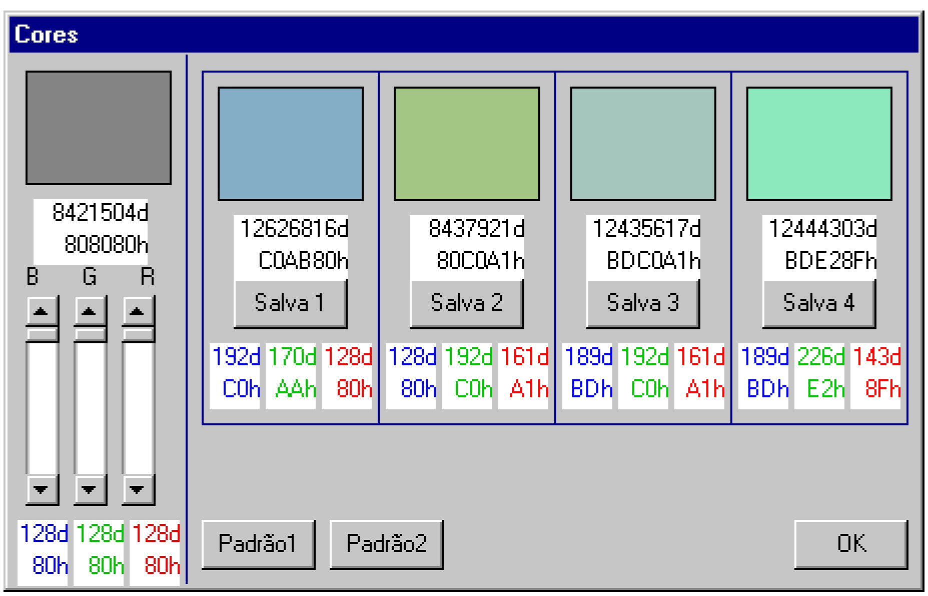


Figura 4.39 – Janela de edição de cores.

Pode-se, também, obter os padrões 1 e 2, por meio de dois botões na parte inferior da janela.Abaixo, tem-se a comparação dos diferentes modos de cores para um exemplo de função.

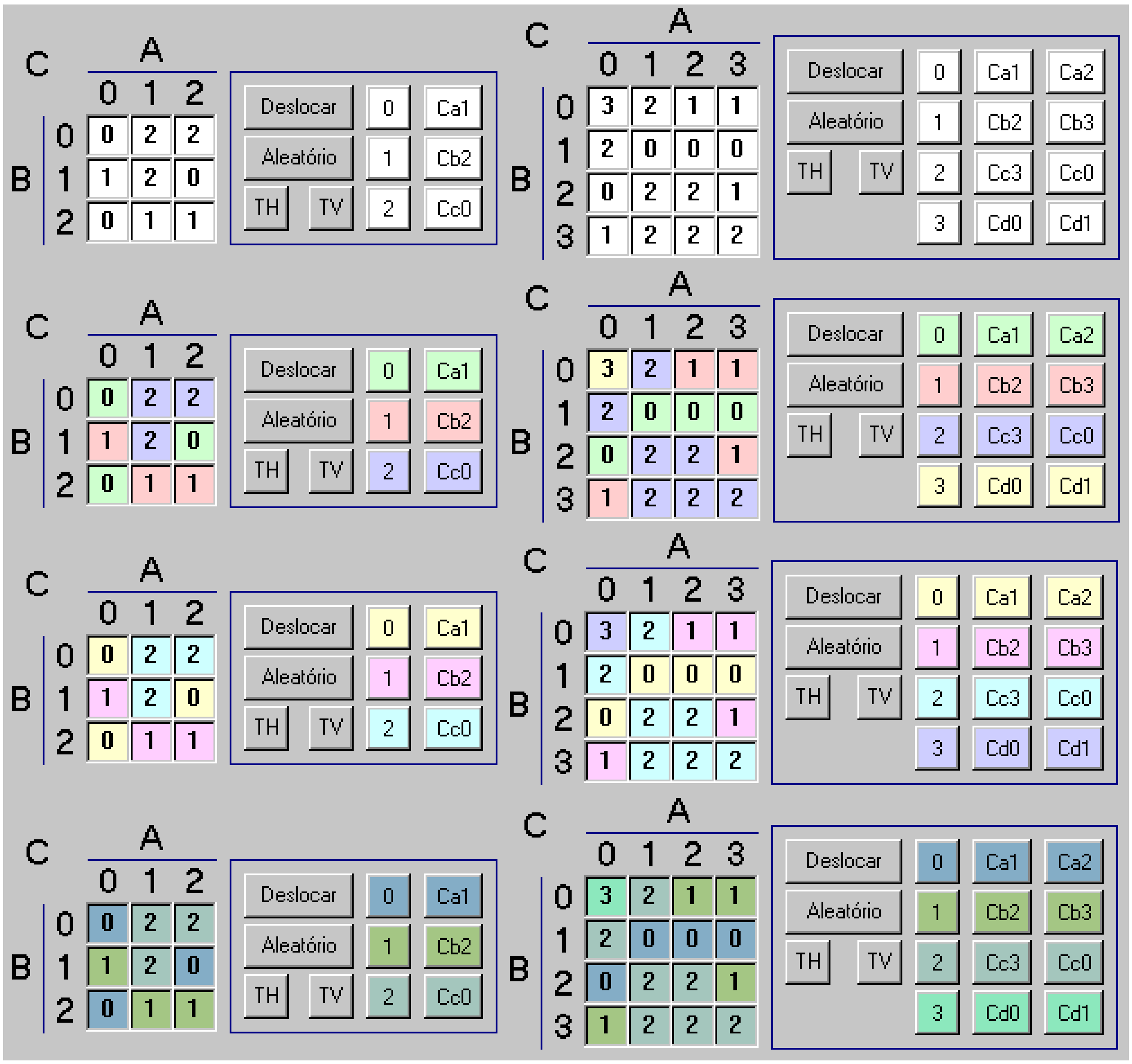


Figura 4.40 – Efeito das cores nos quadros e botões. Sem cores, Padrões 1, 2 e cores do usuário, respectivamente.

4.10 Exemplos de listagem

A seguir, são mostrados exemplos para todas as variações de entradas e valores, sem detalhes.

```
Lógica Multivalores 1.0

 2x1 133

   |--|
 A |01|
   |--|
 B |01|
   |--|

 C1 = A
```

```
Lógica Multivalores 1.0

 2x2 133

      |--|
      |A |
    C |01|
 |----+--|
 |B 0 |01|
 |  1 |10|
 |----+--|

 C1 = (A b B) a (/A b /B)
 C2 = (A a /B) b (/A a B)
```

```
Lógica Multivalores 1.0

 2x3 133

      |--|       |--|
      |A |       |A |
  C=0 |01|  C=1 |01|
 |----+--| |----+--|
 |B 0 |01| |B 0 |10|
 |  1 |10| |  1 |01|
 |----+--| |----+--|

 D1 = (A b B b C) a (/A b /B b C) a (/A b B b /C) a (A b /B b /C)
 D2 = (A a /B a /C) b (/A a B a /C) b (/A a /B a C) b (A a B a C)
```

```
Lógica Multivalores 1.0

 3x1 133

   |---|
 A |012|
   |---|
 B |012|
   |---|

B1 = ( A a 1 ) c ( /A b A )
B2 = ( A b 2 ) a ( /A c A )
B3 = ( A c 0 ) b ( A a /A )
```

```
Lógica Multivalores 1.0

 3x2 133

      |---|
      | A |
    C |012|
 |----+---|
 |  0 |012|
 |B 1 |120|
 |  2 |201|
 |----+---|

C1 = [ (A b B) a (/A b B/) a (A/ b /B) ] c [ (/A c /B) b (A c /B) b (/A
c B) b (A/ c B) b (A c B/) b (A/ c B/) ]
```

```
Lógica Multivalores 1.0

 4x1 121

   |----|
 A |0123|
   |----|
 B |0123|
   |----|

B1 = [ ( A a 1 ) c ( /A b A b 2 ) ] d ( //A c /A c A )
B2 = [ ( A b 2 ) d ( /A c A c 3 ) ] a ( //A d /A d A )
B3 = [ ( A c 3 ) a ( /A d A d 0 ) ] b ( A a //A a /A )
B4 = [ ( A d 0 ) b ( A a /A a 1 ) ] c ( /A b A b //A )
```

```
Lógica Multivalores 1.0

 4x2 033

      |----|
      | A  |
    C |0123|
 |----+----|
 |  0 |0123|
 |B 1 |1230|
 |  2 |2301|
 |  3 |3012|
 |----+----|

C1 = { [ (A b B) a (/A b B/) a (//A b //B) a (A/ b /B) ] c [ (/A c /B)
b (A c /B) b (/A c B) b (//A c B) b (A/ c B/) b (//A c B/) b (A c //B)
b (A/ c //B) ] } d [ (//A d //B) c (/A d //B) c (A d //B) c (//A d /B)
c (/A d /B) c (A/ d /B) c (//A d B) c (A d B) c (A/ d B) c (/A d B/) c
(A d B/) c (A/ d B/) ]
```

A seguir, é mostrado um exemplo para 4x2 com detalhes.

```
Lógica Multivalores 1.0

 4x2 030

      |----|
      | A  |
    C |0123|
 |----+----|
 |  0 |0123|
 |B 1 |1230|
 |  2 |2301|
 |  3 |3012|
 |----+----|

C1 = { [ (A b B) a (/A b B/) a (//A b //B) a (A/ b /B) ] c [ (/A c /B)
b (A c /B) b (/A c B) b (//A c B) b (A/ c B/) b (//A c B/) b (A c //B)
b (A/ c //B) ] } d [ (//A d //B) c (/A d //B) c (A d //B) c (//A d /B)
c (/A d /B) c (A/ d /B) c (//A d B) c (A d B) c (A/ d B) c (/A d B/) c
(A d B/) c (A/ d B/) ]
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
-----------------------------------------------------------------------
----------------------


 ***Exposição da função:***

 4 Valores: 0123


====================================
```

```
      |----|      |----|      |----|
      | A  |      | A  |      | A  |
  c0**|0123|   c1*|0123|   c2 |0123|
 |----+----| |----+----| |----+----|
 |  0 |0111| |  0 |1122| |  0 |2223|
 |B 1 |1110| |B 1 |1221| |B 1 |2232|
 |  2 |1101| |  2 |2211| |  2 |2322|
 |  3 |1011| |  3 |2112| |  3 |3222|
 |----+----| |----+----| |----+----|


Subfunções
c0** = (A b B) a (/A b B/) a (//A b //B) a (A/ b /B)
c1*  = (/A c /B) b (A c /B) b (/A c B) b (//A c B) b (A/ c B/) b (//A c
B/) b (A c //B) b (A/ c //B)
c2   = (//A d //B) c (/A d //B) c (A d //B) c (//A d /B) c (/A d /B) c
(A/ d /B) c (//A d B) c (A d B) c (A/ d B) c (/A d B/) c (A d B/) c (A/
d B/)

Função
C1 = ( c0** c c1* ) d c2
C1 = { [ (A b B) a (/A b B/) a (//A b //B) a (A/ b /B) ] c [ (/A c /B)
b (A c /B) b (/A c B) b (//A c B) b (A/ c B/) b (//A c B/) b (A c //B)
b (A/ c //B) ] } d [ (//A d //B) c (/A d //B) c (A d //B) c (//A d /B)
c (/A d /B) c (A/ d /B) c (//A d B) c (A d B) c (A/ d B) c (/A d B/) c
(A d B/) c (A/ d B/) ]

====================================

      |----|      |----|      |----|
      | A  |      | A  |      | A  |
  c1**|0123|   c2*|0123|   c3 |0123|
 |----+----| |----+----| |----+----|
 |  0 |2122| |  0 |3223| |  0 |0333|
 |B 1 |1222| |B 1 |2233| |B 1 |3330|
 |  2 |2221| |  2 |2332| |  2 |3303|
 |  3 |2212| |  3 |3322| |  3 |3033|
 |----+----| |----+----| |----+----|


Subfunções
c1** = (A c /B) b (/A c B) b (//A c B/) b (A/ c //B)
c2*  = (/A d //B) c (A d //B) c (//A d /B) c (/A d /B) c (//A d B) c
(A/ d B) c (A d B/) c (A/ d B/)
c3   = (//A a B/) d (/A a B/) d (A a B/) d (A/ a //B) d (//A a //B) d
(/A a //B) d (A/ a /B) d (//A a /B) d (A a /B) d (A/ a B) d (/A a B) d
(A a B)

Função
C2 = ( c1** d c2* ) a c3
C2 = { [ (A c /B) b (/A c B) b (//A c B/) b (A/ c //B) ] d [ (/A d //B)
c (A d //B) c (//A d /B) c (/A d /B) c (//A d B) c (A/ d B) c (A d B/)
c (A/ d B/) ] } a [ (//A a B/) d (/A a B/) d (A a B/) d (A/ a //B) d
(//A a //B) d (/A a //B) d (A/ a /B) d (//A a /B) d (A a /B) d (A/ a B)
d (/A a B) d (A a B) ]
```

```
====================================

      |----|      |----|      |----|
      | A  |      | A  |      | A  |
  c2**|0123|   c3*|0123|   c0 |0123|
 |----+----| |----+----| |----+----|
 |  0 |3323| |  0 |0033| |  0 |0100|
 |B 1 |3233| |B 1 |0330| |B 1 |1000|
 |  2 |2333| |  2 |3300| |  2 |0001|
 |  3 |3332| |  3 |3003| |  3 |0010|
 |----+----| |----+----| |----+----|


Subfunções
c2** = (A d //B) c (/A d /B) c (//A d B) c (A/ d B/)
c3*  = (/A a B/) d (A a B/) d (//A a //B) d (/A a //B) d (A/ a /B) d
(//A a /B) d (A/ a B) d (A a B)
c0   = (A b B) a (//A b B) a (/A b B) a (A/ b B/) a (//A b B/) a (/A b
B/) a (A b //B) a (A/ b //B) a (//A b //B) a (A b /B) a (A/ b /B) a (/A
b /B)

Função
C3 = ( c2** a c3* ) b c0
C3 = { [ (A d //B) c (/A d /B) c (//A d B) c (A/ d B/) ] a [ (/A a B/)
d (A a B/) d (//A a //B) d (/A a //B) d (A/ a /B) d (//A a /B) d (A/ a
B) d (A a B) ] } b [ (A b B) a (//A b B) a (/A b B) a (A/ b B/) a (//A
b B/) a (/A b B/) a (A b //B) a (A/ b //B) a (//A b //B) a (A b /B) a
(A/ b /B) a (/A b /B) ]

====================================

      |----|      |----|      |----|
      | A  |      | A  |      | A  |
  c3**|0123|   c0*|0123|   c1 |0123|
 |----+----| |----+----| |----+----|
 |  0 |0003| |  0 |0110| |  0 |1121|
 |B 1 |0030| |B 1 |1100| |B 1 |1211|
 |  2 |0300| |  2 |1001| |  2 |2111|
 |  3 |3000| |  3 |0011| |  3 |1112|
 |----+----| |----+----| |----+----|


Subfunções
c3** = (A a B/) d (/A a //B) d (//A a /B) d (A/ a B)
c0*  = (A b B) a (/A b B) a (//A b B/) a (/A b B/) a (A/ b //B) a (//A
b //B) a (A b /B) a (A/ b /B)
c1   = (/A c /B) b (A c /B) b (//A c /B) b (/A c B) b (A/ c B) b (//A c
B) b (A c B/) b (A/ c B/) b (//A c B/) b (/A c //B) b (A c //B) b (A/ c
//B)

Função
C4 = ( c3** b c0* ) c c1
C4 = { [ (A a B/) d (/A a //B) d (//A a /B) d (A/ a B) ] b [ (A b B) a
(/A b B) a (//A b B/) a (/A b B/) a (A/ b //B) a (//A b //B) a (A b /B)
a (A/ b /B) ] } c [ (/A c /B) b (A c /B) b (//A c /B) b (/A c B) b (A/
c B) b (//A c B) b (A c B/) b (A/ c B/) b (//A c B/) b (/A c //B) b (A
c //B) b (A/ c //B) ]
```

Capítulo 5
Conclusão

O programa realiza exatamente o procedimento determinado pela técnica de síntese empregada, fornecendo a expressão minimizada de maneira intuitiva e fácil de compreender, permitindo ao usuário, caso deseje, obter mais informações sobre as etapas da síntese.

A execução da síntese é feita em tempo real, ou seja, durante a digitação da tabela, levando apenas algumas frações de segundo.

O volume ocupado em disco é reduzido, sendo sua instalação bastante rápida e fácil.

O aplicativo funciona bem nos ambientes Windows95™, Windows98™, WindowsME™ e Windows2000™ e requer poucos recursos de hardware, podendo ser utilizado em equipamentos antigos, como por exemplo, 80486 100MHz, onde pode ser executado com velocidade satisfatória.

Referências

[01]- Azevedo Jr, J. B. de. TTL/CMOS. São Paulo: 4.a Ed., v1.

[02]- Malinowski, G. Many-Valued Logics, University of Lódz. Clarendon Press – Oxford. 1993

[03]- http://turnbull.mcs.st-and.ac.uk/history/Mathematicians/Boole.html

[04]- http://turnbull.mcs.st-and.ac.uk/history/Mathematicians/De_Morgan.html

[05]- http://turnbull.mcs.st-and.ac.uk/history/Mathematicians/Lukasiewicz.html

[06]- http://turnbull.mcs.st-and.ac.uk/history/Mathematicians/Post.html

[07]- Smith, K. C. The prospects for multivalued logic: A technology and applications view. IEEE transactions on computers, vol. C-30, no. 9, September 1981

[08]- Epstein, G. Frieder, G. Rine, D.C. The development of multiple-valued logic as related to computer science. Computer, Vol 7 (9), 1974

[09]- Nascimento, L. P. Uma Ferramenta Automatizada para Análise e Projeto de Circuitos Digitais Multi-Valores, tese de mestrado. Universidade Estadual de Campinas, 2001.

[10]- Hurst, Stanley L. Multi- Valued Logic – Its Status and Its Future. IEEE transactions on computers, vol. C-33, no. 12, 1984.

[11]- Serran, N.V. Circuitos digitais ternários baseados na álgebra de Post, tese de doutorado. Unicamp, 1996.

[12]- Yacoub, M. N. R. D. Proposta de implementação de uma lógica ternária em tecnologia CMOS. 2000.

[13]- Biazon, A J. Filho, Projeto e Construção de uma Porta Universal CMOS em Lógica Ternária, Tese de Mestrado, Universidade Estadual de Campinas, 2000.

[14]- Tang, Z. Ishizuka, O. A learning Multiple-Valued Logic Network Algebra, Algorithm and Aplications. IEEE Transactions on Computer, Vol. 47 no 2 February 1998.

[15]- Malvino, A. P. Eletrônica Digital. São Paulo: McGraw-Hill, 1.a Ed., v1,1987.

[16]- Thoidis, I., Soudris,D., Karafyllidis, I., Christoforidis, S., Thanailakis, A. Quaternary voltage-mode CMOS circuits for multiple-valued logic. IEE Proc – Circuits Devices Syst. Vol 145, No. 2, April 1998

[17]- Mates, Benson. Elementary Logic. New York: Oxford University Press, second edition, 1972. 237p.

[18]- Kabat, W. C., Wojcik, A. S. On the design of 4-Valued Digital Systems. IEEE transactions on computers, vol. C-30, n. 9, September 1981

[19]- Kenneth C. Smith. Multiple Valued Logic: A tutorial and Appreciation. IEEE Computer, 1988.

[20]- Serran, N.V., Jorge, A. M., Dias, J. A. S. A proposal for the implementation of ternary digital circuits. Microelectronics Journal, vol 28, 1997.

[21]- Jorge, A. Martins, “A Universal CMOS gate for Multi-Valued logic (MVL) circuits” - XV International Conference on Microelectronics and Packaging, Manaus, Brazil, pp. 119-124, Set. 2000.

[22]- Biazon, A. J. Filho, Multi-Valued Half Adder circuit using a CMOS Universal gate - XV International Conference on Microelectronics and Packaging, Manaus, Brazil, pp. 125-129 - Set. 2000.

[23]- Nascimento, L. P. An Automated Tool for Analysis and Design of MVL Digital Circuits - XV SBMicro, Brasilia, Brazil, 2001

[24]- Aitken, Peter. Visual Basic 4 Kit do Explorador. Coriolis Group Books 1996

Apêndice 1
Demonstrações

A1.1 Funções

α	**0**	**1**
0	0	0
1	0	1

β	**1**	**0**
1	1	1
0	1	0

$\overline{\alpha}$	**0**	**1**
0	1	1
1	1	0

$\overline{\beta}$	**1**	**0**
1	0	0
0	0	1

Tabela A1.1 – Dois valores.

α	**0**	**1**	**2**
0	0	0	0
1	0	1	1
2	0	1	2

β	**1**	**2**	**0**
1	1	1	1
2	1	2	2
0	1	2	0

γ	**2**	**0**	**1**
2	2	2	2
0	2	0	0
1	2	0	1

$\overline{\alpha}$	**0**	**1**	**2**
0	1	1	1
1	1	2	2
2	1	2	0

$\overline{\beta}$	**1**	**2**	**0**
1	2	2	2
2	2	0	0
0	2	0	1

$\overline{\gamma}$	**2**	**0**	**1**
2	0	0	0
0	0	1	1
1	0	1	2

$\underline{\alpha}$	**0**	**1**	**2**
0	2	2	2
1	2	0	0
2	2	0	1

$\underline{\beta}$	**1**	**2**	**0**
1	0	0	0
2	0	1	1
0	0	1	2

$\underline{\gamma}$	**2**	**0**	**1**
2	1	1	1
0	1	2	2
1	1	2	0

Tabela A1.2 – Três valores.

α	**0**	**1**	**2**	**3**
0	0	0	0	0
1	0	1	1	1
2	0	1	2	2
3	0	1	2	3

β	**1**	**2**	**3**	**0**
1	1	1	1	1
2	1	2	2	2
3	1	2	3	3
0	1	2	3	0

γ	**2**	**3**	**0**	**1**
2	2	2	2	2
3	2	3	3	3
0	2	3	0	0
1	2	3	0	1

δ	**3**	**0**	**1**	**2**
3	3	3	3	3
0	3	0	0	0
1	3	0	1	1
2	3	0	1	2

$\overline{\alpha}$	**0**	**1**	**2**	**3**
0	1	1	1	1
1	1	2	2	2
2	1	2	3	3
3	1	2	3	0

$\overline{\beta}$	**1**	**2**	**3**	**0**
1	2	2	2	2
2	2	3	3	3
3	2	3	0	0
0	2	3	0	1

$\overline{\gamma}$	**2**	**3**	**0**	**1**
2	3	3	3	3
3	3	0	0	0
0	3	0	1	1
1	3	0	1	2

$\overline{\delta}$	**3**	**0**	**1**	**2**
3	0	0	0	0
0	0	1	1	1
1	0	1	2	2
2	0	1	2	3

$\overline{\overline{\alpha}}$	**0**	**1**	**2**	**3**
0	2	2	2	2
1	2	3	3	3
2	2	3	0	0
3	2	3	0	1

$\overline{\overline{\beta}}$	**1**	**2**	**3**	**0**
1	3	3	3	3
2	3	0	0	0
3	3	0	1	1
0	3	0	1	2

$\overline{\overline{\gamma}}$	**2**	**3**	**0**	**1**
2	0	0	0	0
3	0	1	1	1
0	0	1	2	2
1	0	1	2	3

$\overline{\overline{\delta}}$	**3**	**0**	**1**	**2**
3	1	1	1	1
0	1	2	2	2
1	1	2	3	3
2	1	2	3	0

$\underline{\alpha}$	**0**	**1**	**2**	**3**
0	3	3	3	3
1	3	0	0	0
2	3	0	1	1
3	3	0	1	2

$\underline{\beta}$	**1**	**2**	**3**	**0**
1	0	0	0	0
2	0	1	1	1
3	0	1	2	2
0	0	1	2	3

$\underline{\gamma}$	**2**	**3**	**0**	**1**
2	1	1	1	1
3	1	2	2	2
0	1	2	3	3
1	1	2	3	0

$\underline{\delta}$	**3**	**0**	**1**	**2**
3	2	2	2	2
0	2	3	3	3
1	2	3	0	0
2	2	3	0	1

Tabela A1.3 – Quatro valores.

A1.2 Extensões da lei de De Morgan

$\overline{A\alpha B}$

α	**0**	**1**
0	1	1
1	1	0

=

$\overline{A}\beta\overline{B}$

β	**1**	**0**
1	1	1
0	1	0

$\overline{A\beta B}$

β	**0**	**1**
0	1	0
1	0	0

=

$\overline{A}\alpha\overline{B}$

α	**1**	**0**
1	1	0
0	0	0

Tabela A1.4 – Dois valores.

$\overline{A\alpha B}$

$\overline{\alpha}$	**0**	**1**	**2**
0	1	1	1
1	1	2	2
2	1	2	0

=

$\overline{A}\beta\overline{B}$

β	**1**	**2**	**0**
1	1	1	1
2	1	2	2
0	1	2	0

$\underline{A\alpha B}$

$\underline{\alpha}$	**0**	**1**	**2**
0	2	2	2
1	2	0	0
2	2	0	1

=

$\underline{A}\gamma\underline{B}$

γ	**2**	**0**	**1**
2	2	2	2
0	2	0	0
1	2	0	1

$\overline{A\beta B}$

$\overline{\beta}$	**0**	**1**	**2**
0	1	2	0
1	2	2	2
2	0	2	0

=

$\overline{A}\gamma\overline{B}$

γ	**1**	**2**	**0**
1	1	2	0
2	2	2	2
0	0	2	0

$\underline{A\beta B}$

$\underline{\beta}$	**0**	**1**	**2**
0	2	0	1
1	0	0	0
2	1	0	1

=

$\underline{A}\alpha\underline{B}$

α	**2**	**0**	**1**
2	2	0	1
0	0	0	0
1	1	0	1

$\overline{A\gamma B}$

$\overline{\gamma}$	**0**	**1**	**2**
0	1	1	0
1	1	2	0
2	0	0	0

=

$\overline{A}\alpha\overline{B}$

α	**1**	**2**	**0**
1	1	1	0
2	1	2	0
0	0	0	0

$\underline{A\gamma B}$

$\underline{\gamma}$	**0**	**1**	**2**
0	2	2	1
1	2	0	1
2	1	1	1

=

$\underline{A}\beta\underline{B}$

β	**2**	**0**	**1**
2	2	2	1
0	2	0	1
1	1	1	1

Tabela A1.5 – Três valores.

$\overline{A\alpha B} = \overline{A}\beta\overline{B}$

$\overline{\alpha}$	**0**	**1**	**2**	**3**
0	1	1	1	1
1	1	2	2	2
2	1	2	3	3
3	1	2	3	0

=

β	**1**	**2**	**3**	**0**
1	1	1	1	1
2	1	2	2	2
3	1	2	3	3
0	1	2	3	0

$\underline{A\alpha B} = \underline{A}\delta\underline{B}$

$\underline{\alpha}$	**0**	**1**	**2**	**3**
0	3	3	3	3
1	3	0	0	0
2	3	0	1	1
3	3	0	1	2

=

δ	**3**	**0**	**1**	**2**
3	3	3	3	3
0	3	0	0	0
1	3	0	1	1
2	3	0	1	2

$\overline{A\beta B} = \overline{A}\gamma\overline{B}$

$\overline{\beta}$	**0**	**1**	**2**	**3**
0	1	2	3	0
1	2	2	2	2
2	3	2	3	3
3	0	2	3	0

=

γ	**1**	**2**	**3**	**0**
1	1	2	3	0
2	2	2	2	2
3	3	2	3	3
0	0	2	3	0

$\underline{A\beta B} = \underline{A}\alpha\underline{B}$

$\underline{\beta}$	**0**	**1**	**2**	**3**
0	3	0	1	2
1	0	0	0	0
2	1	0	1	1
3	2	0	1	2

=

α	**3**	**0**	**1**	**2**
3	3	0	1	2
0	0	0	0	0
1	1	0	1	1
2	2	0	1	2

$\overline{A\gamma B} = \overline{A}\delta\overline{B}$

$\overline{\gamma}$	**0**	**1**	**2**	**3**
0	1	1	3	0
1	1	2	3	0
2	3	3	3	3
3	0	0	3	0

=

δ	**1**	**2**	**3**	**0**
1	1	1	3	0
2	1	2	3	0
3	3	3	3	3
0	0	0	3	0

$\underline{A\gamma B} = \underline{A}\beta\underline{B}$

$\underline{\gamma}$	**0**	**1**	**2**	**3**
0	3	3	1	2
1	3	0	1	2
2	1	1	1	1
3	2	2	1	2

=

β	**3**	**0**	**1**	**2**
3	3	3	1	2
0	3	0	1	2
1	1	1	1	1
2	2	2	1	2

$\overline{A\delta B} = \overline{A}\alpha\overline{B}$

$\overline{\delta}$	**0**	**1**	**2**	**3**
0	1	1	1	0
1	1	2	2	0
2	1	2	3	0
3	0	0	0	0

=

α	**1**	**2**	**3**	**0**
1	1	1	1	0
2	1	2	2	0
3	1	2	3	0
0	0	0	0	0

$\underline{A\delta B} = \underline{A}\gamma\underline{B}$

$\underline{\delta}$	**0**	**1**	**2**	**3**
0	3	3	3	2
1	3	0	0	2
2	3	0	1	2
3	2	2	2	2

=

γ	**3**	**0**	**1**	**2**
3	3	3	3	2
0	3	0	0	2
1	3	0	1	2
2	2	2	2	2

$\overline{\overline{A\alpha B}} = \overline{\overline{A}}\gamma\overline{\overline{B}}$

$\overline{\overline{\alpha}}$	**0**	**1**	**2**	**3**
0	2	2	2	2
1	2	3	3	3
2	2	3	0	0
3	2	3	0	1

=

γ	**2**	**3**	**0**	**1**
2	2	2	2	2
3	2	3	3	3
0	2	3	0	0
1	2	3	0	1

$\overline{\overline{A\beta B}} = \overline{\overline{A}}\delta\overline{\overline{B}}$

$\overline{\overline{\beta}}$	**0**	**1**	**2**	**3**
0	2	3	0	1
1	3	3	3	3
2	0	3	0	0
3	1	3	0	1

=

δ	**2**	**3**	**0**	**1**
2	2	3	0	1
3	3	3	3	3
0	0	3	0	0
1	1	3	0	1

$\overline{\overline{A\gamma B}} = \overline{\overline{A}}\alpha\overline{\overline{B}}$

$\overline{\overline{\gamma}}$	**0**	**1**	**2**	**3**
0	2	2	0	1
1	2	3	0	1
2	0	0	0	0
3	1	1	0	1

=

α	**2**	**3**	**0**	**1**
2	2	2	0	1
3	2	3	0	1
0	0	0	0	0
1	1	1	0	1

$\overline{\overline{A\delta B}} = \overline{\overline{A}}\beta\overline{\overline{B}}$

$\overline{\overline{\delta}}$	**0**	**1**	**2**	**3**
0	2	2	2	1
1	2	3	3	1
2	2	3	0	1
3	1	1	1	1

=

β	**2**	**3**	**0**	**1**
2	2	2	2	1
3	2	3	3	1
0	2	3	0	1
1	1	1	1	1

Tabela A1.6 – Quatro valores.

$$\neg(A \wedge B) = (\neg A) \vee (\neg B)$$

$\neg\wedge$	**0**	**1**	**2**	**3**		$\vee$	**3**	**2**	**1**	**0**
0	3	3	3	3	=	**3**	3	3	3	3
1	3	2	2	2		**2**	3	2	2	2
2	3	2	1	1		**1**	3	2	1	1
3	3	2	1	0		**0**	3	2	1	0

$$\neg(A \vee B) = (\neg A) \wedge (\neg B)$$

$\neg\vee$	**0**	**1**	**2**	**3**		$\wedge$	**3**	**2**	**1**	**0**
0	3	2	1	0	=	**3**	3	2	1	0
1	2	2	1	0		**2**	2	2	1	0
2	1	1	1	0		**1**	1	1	1	0
3	0	0	0	0		**0**	0	0	0	0

Tabela A1.7 – Lukasiewicz.

A1.3 Conversão para binário e ternário

$A\alpha 1$

0	1	2
1	1	1
0	1	1

$A\beta 2$

0	1	2
2	2	2
2	1	2

$A\gamma 0$

0	1	2
0	0	0
0	0	2

Tabela A1.8 – Dois valores.

$A\alpha 1$

0	1	2	3
1	1	1	1
0	1	1	1

$A\beta 2$

0	1	2	3
2	2	2	2
2	1	2	2

$A\gamma 3$

0	1	2	3
3	3	3	3
3	3	2	3

$A\delta 0$

0	1	2	3
0	0	0	0
0	0	0	3

Tabela A1.9 – Três valores.

$A\alpha 2$

0	1	2	3
2	2	2	2
0	1	2	2

$A\beta 3$

0	1	2	3
3	3	3	3
3	1	2	3

$A\gamma 0$

0	1	2	3
0	0	0	0
0	0	2	3

$A\delta 1$

0	1	2	3
1	1	1	1
0	1	1	3

Tabela A1.10 – Quatro valores.

A1.4 Termos

A1.4.1- Dois valores

c0 - Zeros

0	*A = 0*	*A = 1*
B = 0	$A\beta B$	$\overline{A}\beta B$
B = 1	$A\beta\overline{B}$	$\overline{A}\beta\overline{B}$

c1 – Uns

1	*A = 0*	*A = 1*
B = 0	$\overline{A}\gamma\overline{B}$	$A\gamma\overline{B}$
B = 1	$\overline{A}\gamma B$	$A\gamma B$

Tabela A1.11 – Dois valores.

$$0=(A\beta B)\alpha(\overline{A}\beta B)\alpha(A\beta\overline{B})\alpha(\overline{A}\beta\overline{B})$$
$$1=(\overline{A}\alpha\overline{B})\beta(A\alpha\overline{B})\beta(\overline{A}\alpha B)\beta(A\alpha B)$$

Verificação pelo modo exaustivo:

A=0; B=0	$(0\beta 0)\alpha(1\beta 0)\alpha(0\beta 1)\alpha(1\beta 1)=0\alpha 1\alpha 1\alpha 1=0$ $(1\alpha 1)\beta(0\alpha 1)\beta(1\alpha 0)\beta(0\alpha 0)=1\beta 0\beta 0\beta 0=1$
A=1; B=0	$(1\beta 0)\alpha(0\beta 0)\alpha(1\beta 1)\alpha(0\beta 1)=1\alpha 0\alpha 1\alpha 1=0$ $(0\alpha 1)\beta(1\alpha 1)\beta(0\alpha 0)\beta(1\alpha 0)=0\beta 1\beta 0\beta 0=1$
A=0; B=1	$(0\beta 1)\alpha(1\beta 1)\alpha(0\beta 0)\alpha(1\beta 0)=1\alpha 1\alpha 0\alpha 1=0$ $(1\alpha 0)\beta(0\alpha 0)\beta(1\alpha 1)\beta(0\alpha 1)=0\beta 0\beta 1\beta 0=1$
A=1; B=1	$(1\beta 1)\alpha(0\beta 1)\alpha(1\beta 0)\alpha(0\beta 0)=1\alpha 1\alpha 1\alpha 0=0$ $(0\alpha 0)\beta(1\alpha 0)\beta(0\alpha 1)\beta(1\alpha 1)=0\beta 0\beta 0\beta 1=1$

Verificação por simplificação (duas maneiras):

$$\begin{aligned}
0 &= (A\beta B)\alpha(\overline{A}\beta B)\alpha(A\beta\overline{B})\alpha(\overline{A}\beta\overline{B})\\
0 &= [(A\alpha\overline{A})\beta B]\alpha[(A\alpha\overline{A})\beta\overline{B}]\\
0 &= [0\beta B]\alpha[0\beta\overline{B}]\\
0 &= (B\alpha\overline{B})\beta 0\\
0 &= 0\beta 0\\
0 &= 0
\end{aligned}$$

$$\begin{aligned}
0 &= (A\beta B)\alpha(A\beta\overline{B})\alpha(\overline{A}\beta B)\alpha(\overline{A}\beta\overline{B})\\
0 &= [A\beta(B\alpha\overline{B})]\alpha[\overline{A}\beta(B\alpha\overline{B})]\\
0 &= [A\beta 0]\alpha[\overline{A}\beta 0]\\
0 &= (A\alpha\overline{A})\beta 0\\
0 &= 0\beta 0\\
0 &= 0
\end{aligned}$$

$$\begin{aligned}
1 &= (\overline{A}\alpha\overline{B})\beta(A\alpha\overline{B})\beta(\overline{A}\alpha B)\beta(A\alpha B)\\
1 &= [(\overline{A}\beta A)\alpha\overline{B}]\beta[(\overline{A}\beta A)\alpha B]\\
1 &= [1\alpha\overline{B}]\beta[1\alpha B]\\
1 &= (\overline{B}\beta B)\alpha 1\\
1 &= 1\alpha 1\\
1 &= 1
\end{aligned}$$

$$\begin{aligned}
1 &= (\overline{A}\alpha\overline{B})\beta(\overline{A}\alpha B)\beta(A\alpha\overline{B})\beta(A\alpha B)\\
1 &= [(\overline{B}\beta B)\alpha\overline{A}]\beta[(\overline{B}\beta B)\alpha A]\\
1 &= [1\alpha\overline{A}]\beta[1\alpha A]\\
1 &= (\overline{A}\beta A)\alpha 1\\
1 &= 1\alpha 1\\
1 &= 1
\end{aligned}$$

A1.4.2- Três valores

c0 - Zeros

0	*A = 0*	*A = 1*	*A = 2*
B = 0	$A\beta B$	$\underline{A}\beta B$	$\overline{A}\beta B$
B = 1	$A\beta\underline{B}$	$\underline{A}\beta\underline{B}$	$\overline{A}\beta\underline{B}$
B = 2	$A\beta\overline{B}$	$\underline{A}\beta\overline{B}$	$\overline{A}\beta\overline{B}$

c1 – Uns

1	*A = 0*	*A = 1*	*A = 2*
B = 0	$\overline{A}\gamma\overline{B}$	$A\gamma\overline{B}$	$\underline{A}\gamma\overline{B}$
B = 1	$\overline{A}\gamma B$	$A\gamma B$	$\underline{A}\gamma B$
B = 2	$\overline{A}\gamma\underline{B}$	$A\gamma\underline{B}$	$\underline{A}\gamma\underline{B}$

c2 – Dois

2	*A = 0*	*A = 1*	*A = 2*
B = 0	$\underline{A}\alpha\underline{B}$	$\overline{A}\alpha\underline{B}$	$A\alpha\underline{B}$
B = 1	$\underline{A}\alpha\overline{B}$	$\overline{A}\alpha\overline{B}$	$A\alpha\overline{B}$
B = 2	$\underline{A}\alpha B$	$\overline{A}\alpha B$	$A\alpha B$

Tabela A1.12 – Três valores.

$$0 = (A\beta B)\alpha(\underline{A}\beta B)\alpha(\overline{A}\beta B)\alpha(A\beta\underline{B})\alpha(\underline{A}\beta\underline{B})\alpha(\overline{A}\beta\underline{B})\alpha(A\beta\overline{B})\alpha(\underline{A}\beta\overline{B})\alpha(\overline{A}\beta\overline{B})$$

$$1 = (\overline{A}\gamma\overline{B})\beta(A\gamma\overline{B})\beta(\underline{A}\gamma\overline{B})\beta(\overline{A}\gamma B)\beta(A\gamma B)\beta(\underline{A}\gamma B)\beta(\overline{A}\gamma\underline{B})\beta(A\gamma\underline{B})\beta(\underline{A}\gamma\underline{B})$$

$$2 = (\underline{A}\alpha\underline{B})\gamma(\overline{A}\alpha\underline{B})\gamma(A\alpha\underline{B})\gamma(\underline{A}\alpha\overline{B})\gamma(\overline{A}\alpha\overline{B})\gamma(A\alpha\overline{B})\gamma(\underline{A}\alpha B)\gamma(\overline{A}\alpha B)\gamma(A\alpha B)$$

Antes de demonstrar estas fórmulas, será feita a seguinte demonstração, por inspeção:

$$(A\beta B)\alpha(A\beta\overline{B})\alpha(A\beta\underline{B}) = (A\alpha B)\beta(A\alpha\overline{B})\beta(A\alpha\underline{B})$$
$$(A\gamma B)\beta(A\gamma\overline{B})\beta(A\gamma\underline{B}) = (A\beta B)\gamma(A\beta\overline{B})\gamma(A\beta\underline{B})$$
$$(A\alpha B)\gamma(A\alpha\overline{B})\gamma(A\alpha\underline{B}) = (A\gamma B)\beta(A\gamma\overline{B})\beta(A\gamma\underline{B})$$

AB	$A\beta B$	$A\beta\overline{B}$	$A\beta\underline{B}$	$(\ )\alpha(\ )\alpha(\ )$	$A\alpha B$	$A\alpha\overline{B}$	$A\alpha\underline{B}$	$(\ )\beta(\ )\beta(\ )$
00	0	1	2	0	0	0	0	0
01	1	2	0	0	0	0	0	0
02	2	0	1	0	0	0	0	0
10	1	1	1	1	0	1	1	1
11	1	1	1	1	1	1	0	1
12	1	1	1	1	1	0	1	1
20	2	1	2	1	0	1	2	1
21	1	2	2	1	1	2	0	1
22	2	2	1	1	2	0	1	1

AB	$A\gamma B$	$A\gamma\overline{B}$	$A\gamma\underline{B}$	$(\ )\beta(\ )\beta(\ )$	$A\beta B$	$A\beta\overline{B}$	$A\beta\underline{B}$	$(\ )\gamma(\ )\gamma(\ )$
00	0	0	2	2	0	1	2	2
01	0	2	0	2	1	2	0	2
02	2	0	0	2	2	0	1	2
10	0	1	2	1	1	1	1	1
11	1	2	0	1	1	1	1	1
12	2	0	1	1	1	1	1	1
20	2	2	2	2	2	1	2	2
21	2	2	2	2	1	2	2	2
22	2	2	2	2	2	2	1	2

AB	$A\alpha B$	$A\alpha\overline{B}$	$A\alpha\underline{B}$	$(\ )\gamma(\ )\gamma(\ )$	$A\gamma B$	$A\gamma\overline{B}$	$A\gamma\underline{B}$	$(\ )\alpha(\ )\alpha(\ )$
00	0	0	0	0	0	0	2	0
01	0	0	0	0	0	2	0	0
02	0	0	0	0	2	0	0	0
10	0	1	1	0	0	1	2	0
11	1	1	0	0	1	2	0	0
12	1	0	1	0	2	0	1	0
20	0	1	2	2	2	2	2	2
21	1	2	0	2	2	2	2	2
22	2	0	1	2	2	2	2	2

Tabela A1.13

Então, conclui-se que:

$$(A\beta B)\alpha(A\beta\overline{B})\alpha(A\beta\underline{B}) = A\alpha 1$$
$$(A\gamma B)\beta(A\gamma\overline{B})\beta(A\gamma\underline{B}) = A\beta 2$$
$$(A\alpha B)\gamma(A\alpha\overline{B})\gamma(A\alpha\underline{B}) = A\gamma 0$$

Abaixo, é feita a verificação da veracidade daquelas fórmulas por simplificação:

$$0=(A\beta B)\alpha(\underline{A}\beta B)\alpha(\overline{A}\beta B)\alpha(A\beta\underline{B})\alpha(\underline{A}\beta\underline{B})\alpha(\overline{A}\beta\underline{B})\alpha(A\beta\overline{B})\alpha(\underline{A}\beta\overline{B})\alpha(\overline{A}\beta\overline{B})$$
$$0=(B\alpha 1)\alpha(\overline{B}\alpha 1)\alpha(\underline{B}\alpha 1)$$
$$0=(B\alpha\overline{B}\alpha\underline{B})\alpha 1$$
$$0=0\alpha 1$$
$$0=0$$
$$1=(\overline{A}\gamma\overline{B})\beta(A\gamma\overline{B})\beta(\underline{A}\gamma\overline{B})\beta(\overline{A}\gamma B)\beta(A\gamma B)\beta(\underline{A}\gamma B)\beta(\overline{A}\gamma\underline{B})\beta(A\gamma\underline{B})\beta(\underline{A}\gamma\underline{B})$$
$$1=(B\beta 2)\alpha(\overline{B}\beta 2)\alpha(\underline{B}\beta 2)$$
$$1=(B\beta\overline{B}\beta\underline{B})\beta 2$$
$$1=1\beta 2$$
$$1=1$$
$$2=(\underline{A}\alpha\underline{B})\gamma(\overline{A}\alpha\underline{B})\gamma(A\alpha\underline{B})\gamma(\underline{A}\alpha\overline{B})\gamma(\overline{A}\alpha\overline{B})\gamma(A\alpha\overline{B})\gamma(\underline{A}\alpha B)\gamma(\overline{A}\alpha B)\gamma(A\alpha B)$$
$$2=(B\gamma 0)\alpha(\overline{B}\gamma 0)\alpha(\underline{B}\gamma 0)$$
$$2=(B\gamma\overline{B}\gamma\underline{B})\gamma 0$$
$$2=2\gamma 0$$
$$2=2$$

A1.4.3- Quatro valores

c0 - Zeros

0	*A = 0*	*A = 1*	*A = 2*	*A = 3*
B = 0	$A\beta B$	$\underline{A}\beta B$	$\overline{\overline{A}}\beta B$	$\overline{A}\beta B$
B = 1	$A\beta\underline{B}$	$\underline{A}\beta\underline{B}$	$\overline{\overline{A}}\beta\underline{B}$	$\overline{A}\beta\underline{B}$
B = 2	$A\beta\overline{\overline{B}}$	$\underline{A}\beta\overline{\overline{B}}$	$\overline{\overline{A}}\beta\overline{\overline{B}}$	$\overline{A}\beta\overline{\overline{B}}$
B=3	$A\beta\overline{B}$	$\underline{A}\beta\overline{B}$	$\overline{\overline{A}}\beta\overline{B}$	$\overline{A}\beta\overline{B}$

c1 - Uns

1	*A = 0*	*A = 1*	*A = 2*	*A = 3*
B = 0	$\overline{A}\gamma\overline{B}$	$A\gamma\overline{B}$	$\underline{A}\gamma\overline{B}$	$\overline{\overline{A}}\gamma\overline{B}$
B = 1	$\overline{A}\beta B$	$A\lambda B$	$\underline{A}\lambda B$	$\overline{\overline{A}}\beta B$
B = 2	$\overline{A}\gamma\underline{B}$	$A\lambda\underline{B}$	$\underline{A}\lambda\underline{B}$	$\overline{\overline{A}}\lambda\underline{B}$
B=3	$\overline{A}\gamma\overline{\overline{B}}$	$A\lambda\overline{\overline{B}}$	$\underline{A}\lambda\overline{\overline{B}}$	$\overline{\overline{A}}\lambda\overline{\overline{B}}$

c2 - Dois

2	*A = 0*	*A = 1*	*A = 2*	*A = 3*
B = 0	$\overline{\overline{A}}\delta\overline{\overline{B}}$	$\overline{A}\delta\overline{\overline{B}}$	$A\delta\overline{\overline{B}}$	$\underline{A}\delta\overline{\overline{B}}$
B = 1	$\overline{\overline{A}}\delta\overline{B}$	$\overline{A}\delta\overline{B}$	$A\delta\overline{B}$	$\underline{A}\delta\overline{B}$
B = 2	$\overline{\overline{A}}\delta B$	$\overline{A}\delta B$	$A\delta B$	$\underline{A}\delta B$
B=3	$\overline{\overline{A}}\delta\underline{B}$	$\overline{A}\delta\underline{B}$	$A\delta\underline{B}$	$\underline{A}\delta\underline{B}$

c3 - Três

3	*A = 0*	*A = 1*	*A = 2*	*A = 3*
B = 0	$\underline{A}\alpha\underline{B}$	$\overline{\overline{A}}\alpha\underline{B}$	$\overline{A}\alpha\underline{B}$	$A\alpha\underline{B}$
B = 1	$\underline{A}\alpha\overline{\overline{B}}$	$\overline{\overline{A}}\alpha\overline{\overline{B}}$	$\overline{A}\alpha\overline{\overline{B}}$	$A\alpha\overline{\overline{B}}$
B = 2	$\underline{A}\alpha\overline{B}$	$\overline{\overline{A}}\alpha\overline{B}$	$\overline{A}\alpha\overline{B}$	$A\alpha\overline{B}$
B=3	$\underline{A}\alpha B$	$\overline{\overline{A}}\alpha B$	$\overline{A}\alpha B$	$A\alpha B$

Tabela A1.14 – Quatro valores.

A1.5 Código *Gray* para MVL

Apenas uma variável muda em cada transição.

3x2

0	0
0	1
0	2
1	2
1	0
1	1
2	1
2	2
2	0

4x2

0	0
0	1
0	2
0	3
1	3
1	0
1	1
1	2
2	2
2	3
2	0
2	1
3	1
3	2
3	3
3	0

3x3

0	0	0
0	0	1
0	0	2
0	1	2
0	1	0
0	1	1
0	2	1
0	2	2
0	2	0
1	2	0
1	2	1
1	2	2
1	0	2
1	0	0
1	0	1
1	1	1
1	1	2
1	1	0
2	1	0
2	1	1
2	1	2
2	2	2
2	2	0
2	2	1
2	0	1
2	0	2
2	0	0

Tabela A1.15

Outra possibilidade consiste num código espelhado, possível apenas para bases pares.

0	0
0	1
1	1
1	0

Tabela A1.16 – 2x2

0	0	0
0	0	1
0	1	1
0	1	0

1	1	0
1	1	1
1	0	1
1	0	0

Tabela A1.17 – 2x3

0	0
0	1
0	2
0	3
1	3
1	2
1	1
1	0
2	0
2	1
2	2
2	3
3	3
3	2
3	1
3	0

Tabela A1.18 – 4x2

0	0	0
0	0	1
0	0	2
0	0	3
0	1	3
0	1	2
0	1	1
0	1	0
0	2	0
0	2	1
0	2	2
0	2	3
0	3	3
0	3	2
0	3	1
0	3	0

1	3	0
1	3	1
1	3	2
1	3	3
1	2	3
1	2	2
1	2	1
1	2	0
1	1	0
1	1	1
1	1	2
1	1	3
1	0	3
1	0	2
1	0	1
1	0	0

2	0	0
2	0	1
2	0	2
2	0	3
2	1	3
2	1	2
2	1	1
2	1	0
2	2	0
2	2	1
2	2	2
2	2	3
2	3	3
2	3	2
2	3	1
2	3	0

3	3	0
3	3	1
3	3	2
3	3	3
3	2	3
3	2	2
3	2	1
3	2	0
3	1	0
3	1	1
3	1	2
3	1	3
3	0	3
3	0	2
3	0	1
3	0	0

Tabela A1.19 – 4x3

0	0
0	1
0	2
0	3
0	4
0	5

1	5
1	4
1	3
1	2
1	1
1	0

2	0
2	1
2	2
2	3
2	4
2	5

3	5
3	4
3	3
3	2
3	1
3	0

4	0
4	1
4	2
4	3
4	4
4	5

5	5
5	4
5	3
5	2
5	1
5	0

Tabela A1.20 – 6x2

Apêndice 2
Análise *top-down* do software

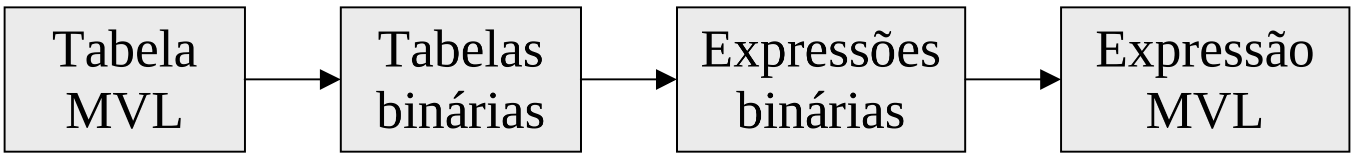


Figura A2.1

Figura A2.2

Obtenção das tabelas binárias, sub-expressões binárias minimizadas e expressões MVL completas.

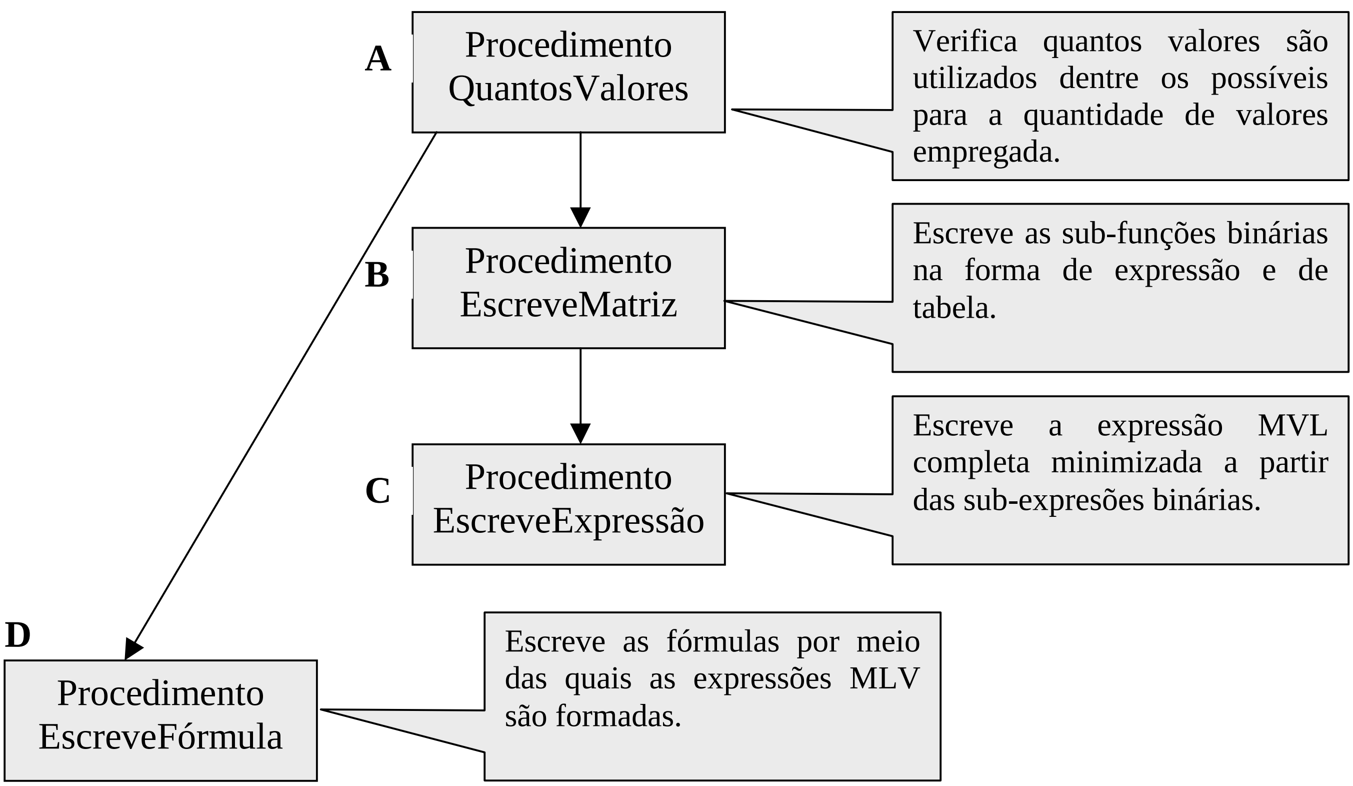


Figura A2.3

Procedimento EscreveMatriz (B)

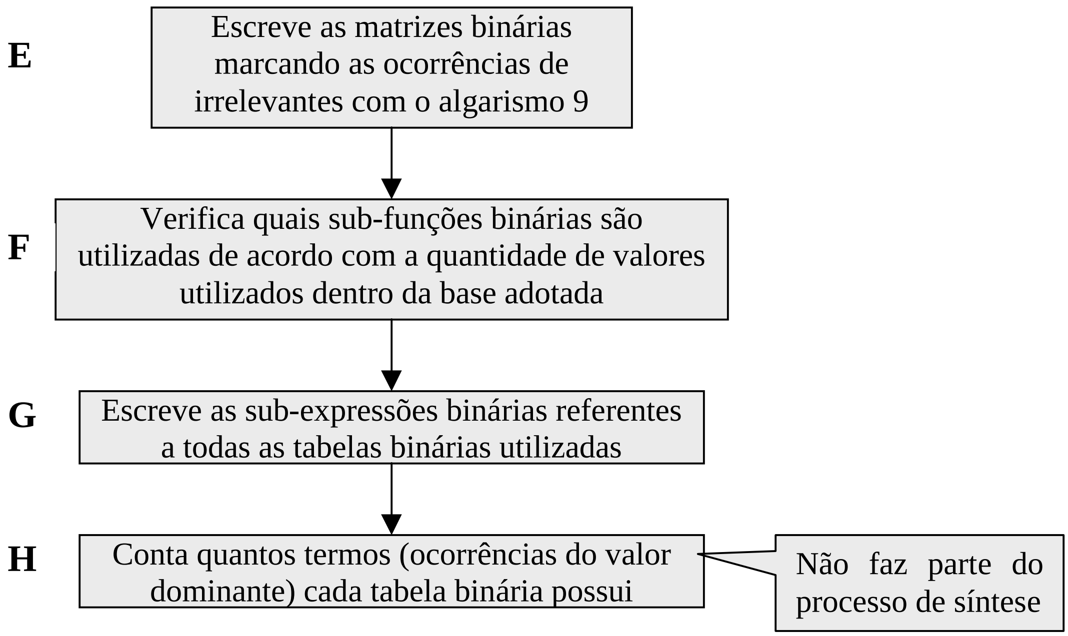


Figura A2.4

Escreve as sub-expressões binárias referentes a todas as tabelas binárias utilizadas (G).

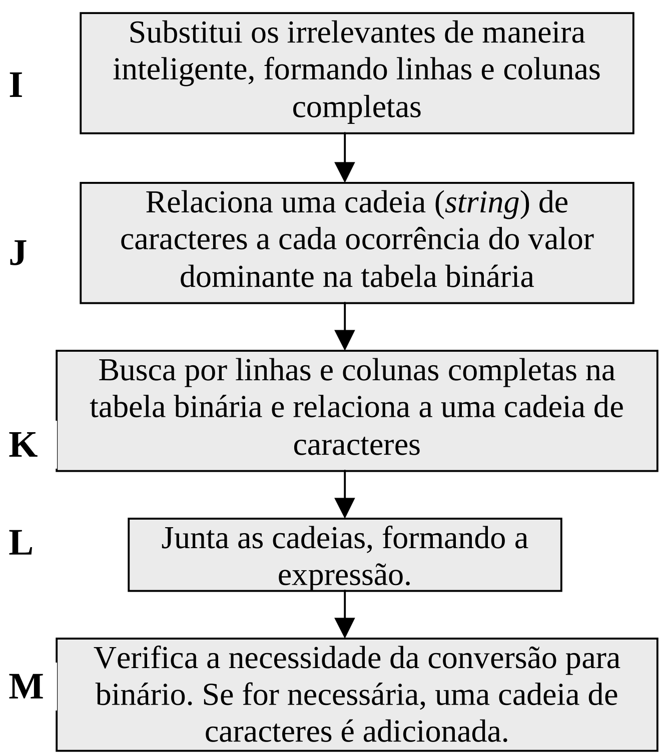


Figura A2.5